



# API HTTP / HTTPS

## Integration Manual

Allmysms.com offers a complete and secure http / https API so you can integrate SMS and MMS in your software applications and websites. This document details the different possibilities offered by the API to manage your SMS mailings, get your answers, your receipts or your blacklist, manage your campaigns and SMS programming, your sub accounts or contact lists in connection with your allmysms.com customer area.



INTEGRATE THE SMS IN  
YOUR APPLICATIONS,  
YOUR WEBSITE

AUTOMATE YOUR TEXT  
OF SHIPMENTS

SETTING SINGLE PLACE  
WITHOUT ADDITIONAL  
COST

VERSION API 9.0

WRITING DATE:28/04/2017

### SAS MS INNOVATIONS

455, Promenade des Anglais  
Nice Premier 06200 NICE

<http://www.allmysms.com>

[support@allmysms.com](mailto:support@allmysms.com)



PRIX APPEL LOCAL

# CONTENTS

1.	DOCUMENT HISTORY	4
2.	API VERSION	5
3.	SECURITY	5
4.	API CONSOLE	5
5.	SIMPLIFIED SENDING SMS	5
6.	SENDING SMS (SENDSMS)	8
7.	SIMULATION OF AN SMS CAMPAIGN (SIMULATECAMPAIGN)	11
8.	DELIVERY REPORTS (GETACKS)	12
9.	RETRIEVING AN DLR BY SMSIDS (GETACKBYSMSID)	14
10.	RECOVERY OF ALL RESPONSES (GETPULLS)	16
11.	RETRIEVING RESPONSES CAMPAIGN (GETPULL)	17
12.	RETRIEVING RESPONSES BY SMSIDS (GETPULLBYSMSID)	19
13.	RETRIEVING ACCOUNT INFORMATION (GETINFO)	21
14.	RECOVERY OF THE BLACKLIST (GETBLACKLIST)	22
15.	DELETE A PROGRAMMED CAMPAIGN (DELETecamPAIGN)	23
16.	CREATE SUB ACCOUNTS (CREATESUBACCOUNT)	24
17.	UPDATE SUB ACCOUNTS (UPDATESUBACCOUNT)	26
18.	RECOVERY OF SUB ACCOUNTS (GETSUBACCOUNTS)	28
19.	CREDITS MANAGEMENT FOR SUB ACCOUNT (MANAGESUBACCOUNTCREDITS)	30
20.	REMOVING SUB ACCOUNT (DELETESUBACCOUNT)	32
21.	CREATE CONTACT LIST (CREATELIST)	32
22.	INSERTING CONTACTS (POPULATELIST)	34
23.	REMOVAL CONTACTS (DELETECONTACTS)	35
24.	DELETING A CONTACT LIST (DETELELIST)	37
25.	RETRIEVING CONTACT LISTS (GETLISTS)	38
26.	RETRIEVING CONTACTS FROM A LIST (GETCONTACTS)	39
27.	RETRIEVING THE NUMBER OF CREDITS PER ISO CODE (GETCREDITSBYCOUNTRYISOCODE)	40
28.	CREATING A SHORTENED URL (SHORTENURL)	41
29.	SENDING EMAIL (SENDEMAIL)	42
30.	SMS BACKOFFICE	45
31.	GLOSSARY	46
32.	NOTES	47

# OBJECTIVE OF THE DOCUMENT

This paper presents how to use and integrate AllMySMS.com HTTP API in your applications and / or websites.

The HTTP API works by transmitting XML or JSON stream input and output to a given URL.

The API works by http and https, but we strongly recommend that you use https to secure exchanges through our Extended Validation SSL certificate.

To send text, variables, XML or JSON feeds containing the information will be transmitted to the platform in GET or POST preferably.

It automatically returns a live return flow containing raw data and XML feed or JSON (identifiers of sendings and receipts).

This paper also presents the protocols to be used, how to encode your stream and restrictions to consider.

It is imperative to read the documentation as a whole prior to interface to the HTTP API AllMySMS.com.

## COSTS

No charge for installation, no subscription fees.

You pay only the price of SMS:

<http://www.allmysms.com/en/sms-pricing/>

## PREREQUISITES

- Open an account [allmysms.com](http://allmysms.com)
- Buy SMS pack or sign a postpaid agreement
- Get your API key on your account: <https://manager.allmysms.com>
- Embed API to your application

## 1. DOCUMENT HISTORY

Document Version	Date	Description
1.17	02/25/2016	<ul style="list-style-type: none"> <li>Adding a sendEmail method for sending bulk emails</li> </ul>
1.16	01/21/2016	<ul style="list-style-type: none"> <li>Adding a getCreditsByCountryIsoCode method to retrieve the number of credits a SMS to a particular country</li> <li>Adding a shortenUrl method to generate a shortened URL of a given Url</li> </ul>
1.15	03/12/2015	<ul style="list-style-type: none"> <li>Adding webhooks, providing the ability to receive push acknowledgments &amp; Answers on a client url configured from the customer area.</li> </ul>
1.14	02/12/2015	<ul style="list-style-type: none"> <li>Updating the getPulls () method, the answers are now recoverable once</li> </ul>
1.13	18/11/2015	<ul style="list-style-type: none"> <li>Adding smsId in the return flow (SendSMS), unique identifiers for each SMS</li> <li>Adding a getAckBySmsId () method for the recovery of a DLR by smsIds</li> <li>Adding a getPullBySmsId () method for retrieving responses smsIds</li> <li>Adding a getPulls () method for the recovery of all answers an account (limited to 1000 by default)</li> </ul>
1.12	04/11/2015	<ul style="list-style-type: none"> <li>Adding a "Document History" section to track changes in this documentation</li> <li><b>SendSMS</b> : Adding creditsUsed, nbSms and nbContacts values to know the credits, SMS sent and the number of relevant contacts</li> <li>Adding a manageSubAccountCredits method to add / delete the master account credit to a sub-account</li> </ul>
1.11	05/12/2015	<ul style="list-style-type: none"> <li><b>createSubAccount</b>: Adding an ACTIVE parameter giving the opportunity to create disabled accounts by default</li> </ul>
1.10	01/24/2015	<ul style="list-style-type: none"> <li>complete overhaul of the documentation</li> </ul>

## 2. API VERSION

The URL of each API method should contain the API version to use. The current version of the API is:

**9.0**

## 3. SECURITY

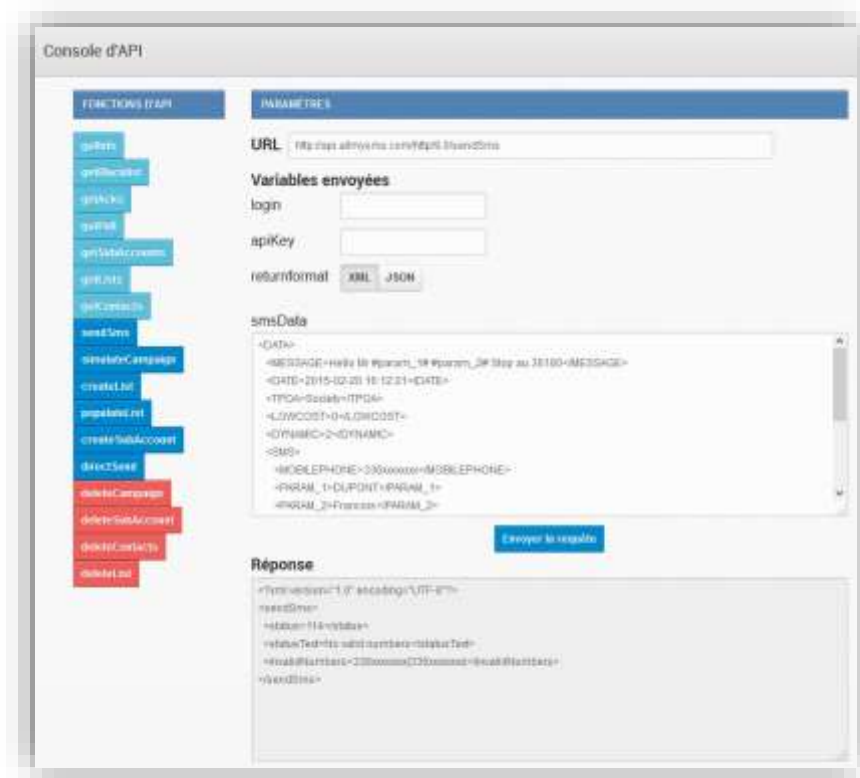
The password of the account used does not pass in your http requests. An API key allows secure exchanges between your system and the allmysms.com API. This key is available on your account:

<https://manager.allmysms.com>

On demand, the API can have a domain name or IP address verification system to provide enhanced security and block any request from an unregistered server. Please contact technical support if this option interests you.

## 4. API CONSOLE

A console is available on your account <https://manager.allmysms.com> to test the API methods. This allows you to check the flow to create as input and the API return flows. This console provides a completely identical behavior to integrate into your application, and your credits will be used, particularly when sending SMS.



## 5. SIMPLIFIED SENDING SMS

## 5.1. OVERVIEW

The URL for simplified sending SMS allows without creating XML or JSON envelope to send SMS by simply waving your settings with GET or POST parameters. This can be useful when you do not need all the features offered by XML and JSON, or to carry out your first tests.

### ADDRESS

<https://api.allmysms.com/http/9.0/>

### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
message	SMS message to send	X
mobile	Mobile number(s) separated by semicolons	X
TPOA	name of the SMS issuer	
dated	date of mailing (to send later)	
campaign	Name of your campaign	

### EXAMPLE OF SIMPLIFIED URL

```
https://api.allmysms.com/http/9.0/?login=monidentifiant&apiKey=maclef&message=Bonjour,%20Merci%20d%27utiliser%20allmysms.com%20STOP%20au%2036180&mobile=0600000000;0611111111&tpoa=allmysms
```

### VARIABLES

List of variables (NAME OF VARIABLE: Definition)

**post:** message to send to the specified list of numbers.

An SMS contains 160 characters. Beyond 160 characters, the technical standard requires that the length of each concatenated SMS passes from 160 to 153 characters, 7 characters are reserved for the UDH header.

Maximum length: 8 SMS or 1224 characters (153 x 8). Please note that passing more than 160 characters, even the first SMS has 153 characters.

For dynamic messages, editable parameters will be of the form PARAM\_1 ##, ## PARAM\_2 ...

**mobile:** phone number in international format (except French numbers) with or without +.

To specify multiple numbers, separated by semicolons. (EG: 0600000000; 0600000001; 0600000002)

**TPOA:**[Optional] message sender (default: 36180).

- Must begin with a letter
- Can only contain alphanumeric characters (a-z0-9) and uppercase letters, or space
- No accented characters or special characters
- 3 characters minimum, 11 maximum

For legal reasons, customizing the issuer for sendings to France will be taken into account if the end of the message contains the STOP au 36180 mention, preceded by a space or a newline.

**campaign:**[Optional] specifies a campaign name for later identification if necessary.

**Date:**[Optional] date in the YYYY-MM-DD HH: MM: SS format in case of delayed transmission programming.

## 5.2. FEED BACK

The response is returned in JSON.

Each API call will return as an answer the status of your request (Status), his narrative explanation (StatusText), your SMS credit (Credits), the numbers that were rejected by the API (invalidNumbers) and the unique identifier of the campaign(campaignId). It allows retrospectively to identify contacts answers.

If an error occurs during the HTTP request, a copy of the error will be sent by email (if requested). If your problems persist, contact support.

### STRUCTURE OF THE RESPONSE

#### 5.2.1.1. EXAMPLE

```
{
  "status": 100,
  "statusText": "Your messages have been sent"
  "invalidNumbers": "",
  "campaignId": "xxxxxxxx"
  "credits" "340"
}
```

## 5.3. WEBHOOK, DELIVERY REPORTS

It is possible to enter a URL for the customer area <https://manager.allmysms.com> on which we will send acknowledgments. (PUSH Method).

### VARIABLES

List of variables sent by GET to your webhook

**smsid:** Unique SMS identifier

**status:** Status of the delivery report, see [Annex acknowledgments](#)

**statusText:** Status in text format

**receptionDate:** Date of operator acknowledgment

**phoneNumber:** Phone number in international format

**campaignId:** Unique identifier of the SMS campaign

## 6. SENDING SMS (SENDSMS)

### 6.1. OVERVIEW

Use the SendSMS API method to achieve your SMS mailings.

#### ADDRESS

<https://api.allmysms.com/http/9.0/sendSms/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
SMSDATA	XML or JSON feed containing the message and phone numbers	X

#### REMARKS

Please use http POST-ended method for sending your parameters (http GET is limited to 1024 characters). For sendings of significant volumes, it is advisable to make multiple requests to the API http by cutting your feed. The maximum runtime of the http protocol might indeed stop it being sent. We recommend that you limit each HTTP call to 2000 maximum contact.

### 6.2. SMSDATA

#### GENERAL STRUCTURE

XML or JSON feed must contain the different variables and values to your SMS sendings.

XML feed example: see [APPENDIX 1: Sample XML SMS sending streams](#)

JSON feed example: see [APPENDIX 2: Example of SMS sending JSON feed](#)

#### VARIABLES

List of variables sent in the XML stream / JSON (VARIABLE NAME: Definition)

**MESSAGE:** message you want to send to the specified list of numbers.

An SMS has 160 characters. Beyond 160 characters, the technical standard requires that the length of each concatenated SMS passes from 160 to 153 characters, 7 characters are reserved for the UDH header.

Maximum length: 8 SMS or 1224 characters (153 x 8). Please note that passing more than 160 characters, even the first SMS has 153 characters.



For dynamic messages, editable parameters will be of the form PARAM\_1 # #, # # PARAM\_2 ... for dynamic, editable parameters will be of the form PARAM\_1 # #, # # PARAM\_2 ...

**DYNAMIC:** indicates the number of variable parameters contained in the message. For example, if the value is 2, the message should contain the strings PARAM\_1 # # and # # PARAM\_2

**CAMPAIGN\_NAME:** [Optional] specifies a campaign name for later identification if necessary.

**DATE:** [Optional] date in the YYYY-MM-DD HH: MM: SS in case of delayed transmission programming.

**TPOA:** [optional] sender of the message (default: 36180).

- Must begin with a letter
- Can only contain alphanumeric characters (a-z0-9) and uppercase letters, or space
- No accented characters or special characters
- 3 characters minimum, 11 maximum

For legal reasons, customizing the issuer for sendings to France will be taken into account if the end of the message contains the STOP sign at 36180, preceded by a space or a newline. Without this mention in the message, the sender will be 36180.

**MAIL\_NOTIF:** [Optional] email notification: 0 to refuse, 1 to accept

**SMS:** beginning of the recipients list

**MOBILEPHONE:** phone number in international format with or without +. Duplicate tag for each phone number.

**PARAM\_X:** dynamic data to insert into the message, X is the parameter number defined in DYNAMIC tag.

**CLIMSGID:** [Optional] identifier provided by the user returned when dlr and answers are sent back

**MASTERACCOUNT:** [Optional] master account login which allows the API to automatically bind a sub-account to a master account

### 6.3. NOTES AND RESTRICTIONS

- encoding:
  - The message must be encoded in UTF-8.
  - The entire SMSDATA parameter must be url-encoded (php: urlencode (); asp: Server.URLEncode ()).
- Special characters:
  - The new line is encoded by '\ n'. Its length counts as 2 characters.
- Formatting numbers

<p>To France, three possible formats:</p> <ul style="list-style-type: none"> <li>▪ 06xxxxxxxx (or 07xxxxxxxx)</li> <li>▪ 336xxxxxxxx (or 337 xxxxxxxx)</li> <li>▪ 336 xxxxxxxx (or 337 xxxxxxxx)</li> </ul>	<p>Outgoing international, two possible formats:</p> <ul style="list-style-type: none"> <li>▪ + CCxxxxxxxx</li> <li>▪ CCxxxxxxxx</li> </ul> <p>Where CC is the country code (Country Code 1 to 3 digits according to destination, ie 32: Belgium, 33: France).</p>
---	--

The lengths are specific to each international operator, there is no length verification for the foreign phone numbers.

Any shipment to a landline number or that do not respect the defined format above will be rejected.

**⚠ Warning :** If the stream is not encoded correctly, messages will be rejected or sent partially.

## 6.4. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

Each API call will return the status of your request (status), its textual explanation (statusText), your SMS credit left (credits), the number of credits used (creditsUsed) and the numbers that were rejected by the API (invalidNumbers).

The unique identifier field (campaignId) will also be returned. It allows retrospectively identify answers contacts.

Each SMS sent to a valid number returns a (smsId) unique.

If an error occurs during the HTTP request, a copy of the error will be sent by email (if requested). If your problems persist, contact support.

### STRUCTURE OF THE RESPONSE

#### 6.4.1.1. EXAMPLE

##### XML FORMAT

```
<sendSMS>
  <status> 100 </ status>
  <statusText> Your messages have been sent</ statusText>
  <invalidNumbers> </ invalidNumbers>
  <campaignId> xxxxxxxx </ campaignId>
  <credits> 370 </ credits>
  <creditsUsed> 15 </ creditsUsed>
  <nbContacts> 1 </ nbContacts>
  <nbSms> 1 </ nbSms>
  <smsIds>
    <sms>
      <phoneNumber> 336XXXXXXXX </ phoneNumber>
      <smsId> f8c964e0-8e0a-11e5-bf51-0025907cd72e </ smsId>
    </ Sms>
  </smsIds>
</ SendSMS>
```

##### JSON FORMAT

```
{
  «status»: 100,
  «statusText»: "Your messages have been sent"
  «invalidNumbers»: "",
  "campaignId": "xxxxxxxxxxxx"
  «credits", "340",
  «creditsUsed": "15"
  "nbContacts": "1"
  "nbSms": "1"
  "smsIds": [
    {
      "phoneNumber": "336XXXXXXXX"
      "smsId": "330792bd-8e0b-11e5-bf51-0025907cd72e"
    }
  ]
}
```

#### 6.4.1.2. VARIABLES RETURN

<b>status:</b>	Statutes of the request (statuses list: cf.APPENDIX 5: Table of Statutes after sending)
<b>statusText:</b>	textual explanation
<b>invalidNumbers:</b>	concatenation of invalid numbers
<b>campaignId:</b>	identification of the SMS campaign carried
<b>credits:</b>	number of remaining credits
<b>creditsUsed:</b>	number of credits used
<b>nbContacts:</b>	number of campaign contacts
<b>nbSms:</b>	number of sent SMS
<b>smsIds:</b>	list of smsIds for each SMS sent
<b>phoneNumber:</b>	SMS phone number
<b>smsId:</b>	Unique SMS identifier

## 6.5. WEBHOOK, RECEPTION OF YOUR RECEIPT OF ACCUSED

It is possible to enter a URL for the customer area <https://manager.allmysms.com> on which we will send acknowledgments. (Method PUSH).

### VARIABLES

List of variables sent by GET to your webhook:

<b>smsId:</b>	Unique SMS identifier
<b>status:</b>	Status of the DLR, see <a href="#">Annex acknowledgments</a>
<b>statusText:</b>	Status text format
<b>receptionDate:</b>	Date of operator acknowledgment
<b>phoneNumber:</b>	Phone number in international format
<b>campaignId:</b>	Unique identifier of the SMS campaign

## 7. SIMULATION OF AN SMS CAMPAIGN (SIMULATECAMPAIGN)

### 7.1. OVERVIEW

Use simulateCampaign method for simulating a campaign to check beforehand the number of contacts, credits, send SMS and invalid numbers sent on request.

### ADDRESS

<https://api.allmysms.com/http/9.0/simulateCampaign/>

### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X

apiKey	API key	X
SMSDATA	XML or JSON feed containing the message and phone numbers (see: <b>Erreur ! Source du renvoi introuvable.</b> )	X

## 7.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

Each simulation will return the status of your request (status), its textual explanation (statusText), the number of credits (nbCredits), the number of contacts (nbContacts), the SMS number (nbSms) and the numbers that were denied by API (invalidNumbers).

### TREE

#### XML FORMAT

```
<simulateCampaign>
  <status> 100 </ status>
  <statusText> Your messages have been sent</ statusText>
  <invalidNumbers> </ invalidNumbers>
  <nbCredits> 45 </ nbCredits>
  <nbContacts> 3 </ nbContacts>
  <nbSms> 3 </ nbSms>
</simulateCampaign>
```

#### JSON FORMAT

```
{
  «status»: 100,
  «statusText»: "Your messages have been sent"
  «invalidNumbers»: "",
  "NbCredits": 45,
  "nbContacts": 2
  "nbSms": 3
}
```

## 8. DELIVERY REPORTS (GETACKS)

### 8.1. OVERVIEW

Use getAcks method to retrieve SMS delivery reports. Mobile operators delivery reports can be upgraded up to 72 hours after shipment.

#### ADDRESS

<https://api.allmysms.com/http/9.0/getAcks/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X

<b>apiKey</b>	API key	X
<b>Campid</b>	campaign ID returned after sending SMS	X
<b>subaccount</b>	Sub account login for which to perform the action	
<b>returnformat</b>	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 8.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```
<getAcks>
  <acks>
    <ack>
      <phoneNumber> 336xxxxxxxx </ phoneNumber>
      <status> 1 </ status>
      <comment> Delivered </ comment>
      <receptionDate> 2014-02-19 11:40:35 </ receptionDate>
    </ack>
    <ack>
      <phoneNumber> 336xxxxxxxx </ phoneNumber>
      <status> 1 </ status>
      <comment> Delivered </ comment>
      <ReceptionDate> 2014-02-19 11:40:36 </ receptionDate>
    </ack>
  </acks>
<cliMsgId> xxxxxx </ cliMsgId>
</ getAcks>
```

#### JSON FORMAT

```
{
  "acks": [
    {
      "phoneNumber": "336xxxxxxxx"
      "status": "1"
      «comment»: "Delivered"
      «receptionDate»: "2014-02-19 11:40:35"
    }
    {
      "phoneNumber": "336xxxxxxxx"
      "status": "1"
      "comment": "Delivered"
      "receptionDate": "2014-02-19 11:40:36"
    }
  ]
  "cliMsgId": "xxxx"
}
```

### VARIABLES RETURN

<b>getAcks</b>	: first tag
<b>ack</b>	: DLR of an SMS. There may be several ACK tags in a stream
<b>status</b>	: Type of DLR received (cf. ANNEX 6: Table of Statutes of accused)
<b>comment</b>	: Code for the detailed explanation of the DLR received (if available)
<b>climsgid</b>	: Optional parameter provided by the customer when sending the request SMS

## OTHER VARIABLES RETURN

Here are other possible system responses:

**error:** An error has occurred. Contains the reason for the error (connection problem, invalid parameters ...)

## 8.3. WEBHOOK, RECEPTION OF YOUR RECEIPT OF ACCUSED

It is possible to enter a URL for the customer area <https://manager.allmysms.com> on which we will send acknowledgments. (Method PUSH).

### VARIABLES

List of variables sent by GET to your webhook

**smsId:** Unique SMS identifier

**status:** Status of the DLR, see [Annex acknowledgments](#)

**statusText:** Status text format

**receptionDate:** Date of operator acknowledgment

**phoneNumber:** Phone number in international format

**campaignId:** Unique identifier of the SMS campaign

## 9. RETRIEVING AN DLR BY SMSIDS (GETACKBYSMSID)

### 9.1. OVERVIEW

Use `getAckBySmsId` method to retrieve the DLR of a given text message. The delivery report of the mobile operators can be upgraded up to 72 hours after shipment.

### ADDRESS

<https://api.allmysms.com/http/9.0/getAckBySmsId/>

### SETTINGS

Parameter	Description	Mandatory
<b>login</b>	allmysms.com customer login	X
<b>apiKey</b>	API key	X
<b>smsIds</b>	SMS ID returned when sending SMS	X
<b>subaccount</b>	Sub account login for which to perform the action	
<b>returnformat</b>	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 9.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```
<getAckBySmsId>
  <phoneNumber> 336XXXXXXXX </ phoneNumber>
  <status> 1 </ status>
  <comment> Delivered </ comment>
  <receptionDate> 2015-11-01 10:55:00 </ receptionDate>
</getAckBySmsId>
```

#### JSON FORMAT

```
{
  "phoneNumber": "336XXXXXXXX"
  "status": "1"
  "comment": "Delivered"
  "receptionDate": "2015-11-01 10:55:00"
}
```

### VARIABLES RETURN

**getAckBySmsId**: receipt of an SMS. There may be several ACK tags in a stream  
**status** : Type of DLR received (cf. ANNEX 6: Table of Statutes of accused)  
**comment** : Code for the detailed explanation of the DLR received (if available)  
**phoneNumber** : SMS mobile phone number  
**receptionDate** : date of receipt

### OTHER VARIABLES RETURN

Instead different ACK, here are other possible system responses:

**error**: An error has occurred. Contains the reason for the error (connection problem, invalid parameters ...)

## 9.3. WEBHOOK, RECEPTION OF YOUR DLR

It is possible to enter a URL for the customer area <https://manager.allmysms.com> on which we will send acknowledgments. (Method PUSH).

### VARIABLES

List of variables sent by GET to your webhook:

**smsId**: Unique SMS identifier  
**status**: Status of the DLR, see [Annex acknowledgments](#)  
**statusText**: Status text format  
**receptionDate**: Date of operator acknowledgment  
**phoneNumber**: Phone number in international format  
**campaignId**: Unique identifier of the SMS campaign

## 10. RECOVERY OF ALL RESPONSES (GETPULLS)

### 10.1. OVERVIEW

Use getPulls method to retrieve the SMS answers (SMS-MO) for all your sendings, from most recent to oldest. The already collected responses will no longer be. You can still access it using getPull () method, stating the campaign ID.

#### ADDRESS

<https://api.allmysms.com/http/9.0/getPulls/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
limit	maximum number of responses to recover (default: 1000)	
subaccount	Sub account login for which to perform the action	
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

### 10.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

#### TREE

##### XML FORMAT

```
<GetPulls>
  <mos>
    <mo>
      <phoneNumber> 336xxxxxxxx </ phoneNumber>
      <id> 32154 </ id>
      <campaignId> ap_284xxxxx </ campaignId>
      <Message> Answer 1 </ message>
      <receptionDate> 2014-02-14 3:09:05 p.m. </ receptionDate>
    </ Mo>
    <mo>
      <phoneNumber> 336xxxxxxxx </ phoneNumber>
      <id> 32157 </ id>
      <campaignId> ap_284xxxxx </ campaignId>
      <Message> Answer 2 </ message>
      <receptionDate> 2014-02-14 3:25:48 p.m. </ receptionDate>
    </ Mo>
  </mos>
</ GetPulls>
```

##### JSON FORMAT

```
{
  "Mos": [
    {
```



```

    <phoneNumber": "336xxxxxxxx"
    "Id": "32154"
    "campaignId": "ap_28423xxxxx"
    "Message": "Answer 1"
    <receptionDate>: "2014-02-14 3:09:05 p.m."
  }
  {
    <phoneNumber": "336xxxxxxxx"
    "Id": "32157"
    "campaignId": "ap_28423xxxxx"
    "Message": "Answer 1"
    <receptionDate>: "2014-02-14 3:25:48 p.m."
  }
]
}

```

## VARIABLES RETURN

**getPulls** : Is the tag of the highest level of the stream.

**mo** : message received.

**phoneNumber**: Phone number of the contact that responded. Allows you to link the number contacted during the campaign and the SMS received in response.

**id** : ID of the response.

**campaignId** : ID of the campaign sent and which contact has responded. Let make the connection between the SMS campaign and the SMS received in response.

**message** : Message from the contact that responded.

**receptionDate**: Date of receipt of the response.

## OTHER VARIABLES RETURN

Instead of different SMS-MO, here are the possible responses of the system:

**error**:An error has occurred. Contains the reason for the error (connection problem, invalid parameters ...)

## 10.3. WEBHOOK, RECEPTION OF YOUR ANSWERS

It is possible to enter a URL for the customer area <https://manager.allmysms.com> on which we will send the answers to your mailings. (Method PUSH).

### VARIABLES

List of variables sent by GET to your webhook

**smsId**: Unique SMS identifier

**smsMold**: Unique identifier of the reply SMS

**receptionDate**: Date of reply

**phoneNumber**: Phone number in international format

**campaignId**: Unique identifier of the SMS campaign

**post**: Message from the SMS response

## 11. RETRIEVING RESPONSES CAMPAIGN (GETPULL)

## 11.1. OVERVIEW

Use getPull method to retrieve the answers SMS (SMS-MO) for your SMS campaigns.

### ADDRESS

<https://api.allmysms.com/http/9.0/getPull/>

### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
Campid	campaign ID returned after sending SMS	X
subaccount	Sub account login for which to perform the action	
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 11.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```
<GetPull>
  <mos>
    <mo>
      <phoneNumber> 336xxxxxxxx </ phoneNumber>
      <id> 32154 </ id>
      <campaignId> ap_284xxxxx </ campaignId>
      <Message> Answer 1 </ message>
      <receptionDate> 2014-02-14 3:09:05 p.m. </ receptionDate>
    </ Mo>
    <mo>
      <phoneNumber> 336xxxxxxxx </ phoneNumber>
      <id> 32157 </ id>
      <campaignId> ap_284xxxxx </ campaignId>
      <Message> Answer 2 </ message>
      <receptionDate> 2014-02-14 3:25:48 p.m. </ receptionDate>
    </ Mo>
  </mos>
  <cliMsgId> xxxx </ cliMsgId>
</ GetPull>
```

#### JSON FORMAT

```
{
  "Mos": [
    {
      "phoneNumber": "336xxxxxxxx"
      "Id": "32154"
      "campaignId": "ap_28423xxxxx"
      "Message": "Answer 1"
    }
  ]
}
```

```

    <<receptionDate>>: "2014-02-14 3:09:05 p.m."
  }
  {
    <<phoneNumber>>: "336xxxxxxxx"
    "Id": "32157"
    "campaignId": "ap_28423xxxxx"
    "Message": "Answer 1"
    <<receptionDate>>: "2014-02-14 3:25:48 p.m."
  }
}
"CliMsgId": "xxxxx"
}

```

## VARIABLES RETURN

- getPull** : Is the tag of the highest level of the stream.
- mo** : message received.
- phoneNumber**: Phone number of the contact that responded. Allows you to link the number contacted during the campaign and the SMS received in response.
- id** : ID of the response.
- campaignId** : ID of the campaign sent and which contact has responded. Let make the connection between the SMS campaign and the SMS received in response.
- message** : Message from the contact that responded.
- receptionDate** : Date of receipt of the response.

## OTHER VARIABLES RETURN

Instead of different SMS-MO, here are the possible responses of the system:

**error**: An error has occurred. Contains the reason for the error (connection problem, invalid parameters ...)

## 11.3. WEBHOOK, RECEPTION OF YOUR ANSWERS

It is possible to enter a URL for the customer area <https://manager.allmysms.com> on which we will send the answers to your mailings. (Method PUSH).

### VARIABLES

List of variables sent by GET to your webhook

- smsId**: Unique SMS identifier
- smsMold**: Unique identifier of the reply SMS
- receptionDate**: Date of reply
- phoneNumber**: Phone number in international format
- campaignId**: Unique identifier of the SMS campaign
- post**: Message from the SMS response

## 12. RETRIEVING RESPONSES BY SMSIDS (GETPULLBYSMSID)

### 12.1. OVERVIEW

Use `getPullBySmsId` method to retrieve the response (SMS-MO) to a given SMS.

## ADDRESS

<https://api.allmysms.com/http/9.0/getPullBySmsId/>

## SETTINGS

Parameter	Description	Mandatory
<code>login</code>	allmysms.com customer login	X
<code>apiKey</code>	API key	X
<code>smsId</code>	SMS ID returned when sending SMS	X
<code>subaccount</code>	Sub account login for which to perform the action	
<code>returnformat</code>	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 12.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

## TREE

### XML FORMAT

```
<Getpullbysmsid>
  <phoneNumber> 336XXXXXXXXX </ phoneNumber>
  <campaignId> XXXXXXXXXXXX </ campaignId>
  <Message> This is an answer </ message>
  <ReceptionDate> 2015-11-01 4:00:00 p.m. </ receptionDate>
</ Getpullbysmsid>
```

### JSON FORMAT

```
{
  "phoneNumber ":" 336XXXXXXXXX "
  "campaignId ":" XXXXXXXXXXXX "
  "message ":" Test response "
  «receptionDate»: "2015-11-02 4:01:00 p.m."
}
```

## VARIABLES RETURN

**getpullbysmsid**: Is the highest level tag of the stream containing the response received.

**phoneNumber**: Phone number of the contact that responded. Allows you to link the number contacted during the campaign and the SMS received in response.

**campaignId** : ID of the campaign sent and which contact has responded. Let make the connection between the SMS campaign and the SMS received in response.

**message** : Message from the contact that responded.

**receptionDate**: Date of receipt of the response.

## OTHER VARIABLES RETURN

Instead of different SMS-MO, here are the possible responses of the system:

**error:** An error has occurred. Contains the reason for the error (connection problem, invalid parameters ...)

## 12.3. WEBHOOK, RECEPTION OF YOUR ANSWERS

It is possible to enter a URL for the customer area <https://manager.allmysms.com> on which we will send the answers to your mailings. (Method PUSH).

### VARIABLES

List of variables sent by GET to your webhook:

**smsId:** Unique SMS identifier

**smsMold:** Unique identifier of the reply SMS

**receptionDate:** Date of reply

**phoneNumber:** Phone number in international format

**campaignId:** Unique identifier of the SMS campaign

**post:** Message from the SMS response

## 13. RETRIEVING ACCOUNT INFORMATION (GETINFO)

### 13.1. OVERVIEW

Check the existence of an account and the number of remaining credit through getInfo method.

#### ADDRESS

<https://api.allmysms.com/http/9.0/getInfo/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
subaccount	Sub account login for which to perform the action	
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

### 13.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

## TREE

### XML FORMAT

```
<GetInfo>
  <status> Active Account </ status>
  <credits> 1847 </ credits>
  <apiKey> vikmakey </ apiKey>
  <lastName> Dupont </ lastName>
  <firstName> Mark </ firstName>
  <Society> allmysms.com </ Society>
  <email> support@allmysms.com </ email>
</ GetInfo>
```

### JSON FORMAT

```
{
  «status»: "Active Account"
  «credits»: "1847"
  "apiKey": "vikmakey"
  "lastName": "Dupont"
  "firstName": "Mark"
  "Society": "allmysms.com"
  "Email", "support@allmysms.com"
}
```

## VARIABLES RETURN

**Status** Status of the requested account

**Credits:** Number of credits remaining on the account (if available)

## 14. RECOVERY OF THE BLACKLIST (GETBLACKLIST)

### 14.1. OVERVIEW

Use getBlacklist method to retrieve your blacklist. It contains all of your campaigns STOP, the IASP and your manual additions.

Note: To delete a number from the blacklist, send START to the short number used when blacklisting from the mobile.

## ADDRESS

<https://api.allmysms.com/http/9.0/getBlacklist/>

## SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
subaccount	Sub account login for which to perform the action	

**returnformat**

desired type of return XML or JSON (if not specified, the return will be JSON)

## 14.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```

<GetBlacklist>
  <blacklist>
    <contact>
      <phoneNumber> 336xxxxxxxx </ phoneNumber>
      <comment> stop </ comment>
      <receptionDate> 2014-02-18 12:10:23 </ receptionDate>
    </ Contact>
    <contact>
      <phoneNumber> 33xxxxxxxx </ phoneNumber>
      <comment> Refused </ comment>
      <receptionDate> 2014-02-18 11:32:11 </ receptionDate>
    </ Contact>
  </ Blacklist>
</ GetBlacklist>

```

#### JSON FORMAT

```

[
  "Blacklist": [
    { «phoneNumber»: "336XXXXXXXX"
      «comment», "stop"
      «receptionDate»: "2014-02-18 12:10:23"
    }
  ]
]

```

### VARIABLES RETURN

- getBlacklist** : Is the tag of the highest level of the stream.
- comment** : Contact type. (Stop: STOP Refused: bounces, unsubscribe manual: manual additions)
- phoneNumber** : Phone number blacklisted
- receptionDate** : To know the date of blacklisting

## 15. DELETE A PROGRAMMED CAMPAIGN (DELETECAMPAIGN)

### 15.1. OVERVIEW

The deleteCampaign method deletes a scheduled campaign. An immediate campaign cannot be deleted.

#### ADDRESS

<https://api.allmysms.com/http/9.0/deleteCampaign/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
Campid	Id of the campaign to delete	X
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 15.2. FEED BACK

The format of the return flow will correspond to that requested when sending, XML or JSON.

### TREE

#### XML FORMAT

```
<DeleteCampaign>
<status> OK </ status>
</ DeleteCampaign>
```

#### JSON FORMAT

```
{
  «status»: "OK"
}
```

### VARIABLES RETURN

**Status** : Return of our api APPENDIX 7: Country suppression status Table for different statuses.

## 16. CREATE SUB ACCOUNTS (CREATESUBACCOUNT)

### 16.1. OVERVIEW

Create sub accounts using the method createSubAccount.

### ADDRESS

<https://api.allmysms.com/http/9.0/createSubAccount/>

### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
accountData	Data in the new report to create	X



## 16.2. ACCOUNTDATA

### GENERAL STRUCTURE

XML or JSON feed must contain the corresponding different variables and values to your SMS.

XML feed example: see APPENDIX 8: Sample XML stream in creation account

Example JSON stream See APPENDIX 9: Example of JSON stream under creation account

### VARIABLES

List of variables sent in the XML stream / JSON (VARIABLE NAME: Definition)

<b>FIRSTNAME</b>	: (Required) Name of account under
<b>LASTNAME</b>	: (Required) sub account Name
<b>SOCIETY</b>	: Company sub account
<b>MOBILE</b>	: Mobile phone sub account
<b>E-MAIL</b>	: (Required) Email sub account
<b>LOGIN</b>	: (Required) sub account Login
<b>PASSWORD</b>	: (Required) Password sub account (minimum 6 characters)
<b>PHONE</b>	: Phone sub account
<b>WEBSITE</b>	: sub account Website
<b>ADDRESS</b>	: sub account Address
<b>Zip</b>	: Postcode sub account
<b>CITY</b>	: City sub account
<b>COUNTRY</b>	: Countries sub account
<b>BILLFIRSTNAME</b>	: First sub account BILLING
<b>BILLLASTNAME</b>	: BILLING sub account Name
<b>BILLSOCIETY</b>	: Society sub account BILLING
<b>BILLADDRESS</b>	: BILLING sub account Address
<b>BILLZIP</b>	: ZIP Code sub account BILLING
<b>BILLCITY</b>	: City sub account BILLING
<b>BILLCOUNTRY</b>	: Countries sub account BILLING
<b>BILLEMAIL</b>	: Email sub account BILLING
<b>BILLMOBILE</b>	: Mobile phone sub account BILLING
<b>BILLPHONE</b>	: Phone sub account BILLING
<b>SENDEMAILTOCUSTOMER</b>	: Sending an email summarizing the sub account (Boolean 1: yes, 0: no)
<b>SENDEMAILTOMASTERACCOUNT</b>	: Sending an email summarizing the master account (Boolean 1: yes, 0: no)
<b>POSTPAID</b>	: Setting the account in post payment (only possible if the master account itself in post payment, Boolean 1: yes, 0: no)
<b>PROFILE</b>	: Type in the account profile (access rights to define MS Innovations)
<b>ACTIVE</b>	: State sub account (Boolean 1: active, 0: inactive by default Active is 1 that is to say active)

## 16.3. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

## TREE

### XML FORMAT

```
<CreateSubAccount>
  <status> 1 </ status>
  <text> Sub-account successfully created </ text>
</ CreateSubAccount>
```

### JSON FORMAT

```
{
  «status»: "1"
  "Text": "Sub-account successfully created"
}
```

## VARIABLES RETURN

- status** Status of the request (see 10 ANNEX: Table of creating returns / modification sub account)
- text** : Description of creation in processing account

## 17. UPDATE SUB ACCOUNTS (UPDATESUBACCOUNT)

### 17.1. OVERVIEW

Change your sub accounts using the method updateSubAccount.

#### ADDRESS

<https://api.allmysms.com/http/9.0/updateSubAccount/>

#### SETTINGS

Parameter	Description	Mandatory
<b>login</b>	allmysms.com customer login	X
<b>apiKey</b>	API key	X
<b>accountData</b>	data under the new account to edit	X
<b>subaccount</b>	Login under the account in question	X

### 17.2. ACCOUNTDATA

#### VARIABLES

List of variables sent in the XML stream / JSON (VARIABLE NAME: Definition) .No variable is mandatory if it is not specified in the stream, the system will keep the previously saved value.

- FIRSTNAME** : First sub account

<b>LASTNAME</b>	: sub account Name
<b>SOCIETY</b>	: Company sub account
<b>MOBILE</b>	: Mobile phone sub account
<b>E-MAIL</b>	: Email sub account
<b>LOGIN</b>	: Login Sub Account
<b>PASSWORD</b>	: Password sub account (minimum 6 characters)
<b>PHONE</b>	: Phone sub account
<b>WEBSITE</b>	: sub account Website
<b>ADDRESS</b>	: sub account Address
<b>Zip</b>	: Postcode sub account
<b>CITY</b>	: City sub account
<b>COUNTRY</b>	: Countries sub account
<b>BILLFIRSTNAME</b>	: First sub account BILLING
<b>BILLLASTNAME</b>	: BILLING sub account Name
<b>BILLSOCIETY</b>	: Society sub account BILLING
<b>BILLADDRESS</b>	: BILLING sub account Address
<b>BILLZIP</b>	: ZIP Code sub account BILLING
<b>BILLCITY</b>	: City sub account BILLING
<b>BILLCOUNTRY</b>	: Countries sub account BILLING
<b>BILLEMAIL</b>	: Email sub account BILLING
<b>BILLMOBILE</b>	: Mobile phone sub account BILLING
<b>BILLPHONE</b>	: Phone sub account BILLING
<b>ACTIVE</b>	: State sub account (Boolean 1: active 0: inactive)

## 17.3. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```
<UpdateSubAccount>
  <status> 1 </ status>
  <text> Sub-account successfully updated </ text>
</ UpdateSubAccount>
```

#### JSON FORMAT

```
{
  «status»: "1"
  "Text": "Sub-account successfully updated"
}
```

### VARIABLES RETURN

- status** : Status of the request (see10 ANNEX: Table of creating returns / modification sub account)
- text** : Description of treatment modification under Account

## 18. RECOVERY OF SUB ACCOUNTS (GETSUBACCOUNTS)

### 18.1. OVERVIEW

List the list of your sub accounts using the method `getSubAccounts`.

#### ADDRESS

<https://api.allmysms.com/http/9.0/getSubAccounts/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

### 18.2. FEED BACK

The format of the response stream will match the one used when sending, XML or JSON.

#### TREE

##### XML FORMAT

```
<GetSubAccounts>
  <total> 3 </ Total>
  <subaccounts>
    <subaccount>
      <Login>subaccountx </ Login>
      <LastName> Doe </ LastName>
      <FirstName> John </ FirstName>
      <ClientType> Company </ ClientType>
      <Company> My Company </ Company>
      <Phone> 33xxxxxxxx </ phone>
      <Mobile> 33xxxxxxxx </ Mobile>
      <Fax> </ fax>
      <Email> subaccountx msinnovations.com @ </ email>
      <WebSite> </ WebSite>
      <Address> 1240 route sinkholes </ Address>
      <Zip> 06560 </ zip>
      <City>Valbonne </ City>
      <Country> France </ Country>
      <CreationDate> 2014-01-01 10:30:00 </ CreationDate>
      <LastVisit> 2014-05-20 2:30:00 p.m. </ LastVisit>
      <LastActivity> 2014-05-22 5:15:00 p.m. </ LastActivity>
      <BillLastName> Doe </ BillLastName>
      <BillFirstName> John </ BillFirstName>
      <BillCompany> My Company </ BillCompany>
      <BillPhone> 33xxxxxxxx </ BillPhone>
      <BillMobile> 33xxxxxxxx </ BillMobile>
      <BillEmail> subaccount1@msinnovations.com </ BillEmail>
      <BillAddress> 1240 road sinkholes </ BillAddress>
      <BillZip> 06560 </ BillZip>
      <BillCity> Valbonne </ BillCity>
      <BillCountry> France </ BillCountry>
      <credits> 1500 </ Credits>
    </subaccount>
  </subaccounts>
</GetSubAccounts>
```

```

        </subaccount>
      </subaccount>
      ...
    </subaccount>
  </subaccounts>
</ GetSubAccounts>

```

## JSON FORMAT

```

{
  "Total": 15,
  "Subaccounts": [
    {
      "Login": "subaccountx"
      "LastName", "Doe"
      "FirstName": "John"
      "ClientType", "Company",
      "Company": "My Company"
      "Phone": "33xxxxxxxxx"
      "Mobile" "33xxxxxxxxx"
      "Fax": "",
      "Email" "subaccountx@msinnovations.com"
      "Address": "1240 road sinkholes"
      "Zip": "06560"
      "City": "Valbonne"
      "Country": "France"
      "CreationDate": "2014-01-01 10:30:00"
      "LastVisit": "2014-05-20 2:30:00 p.m."
      "LastActivity": "2014-05-22 5:15:00 p.m."
      "BillLastName": "Doe"
      "BillFirstName": "John"
      "BillCompany": "My Company"
      "BillPhone": "33xxxxxxxxx"
      "BillMobile": "33xxxxxxxxx"
      "BillEmail": "subaccountx@msinnovations.com"
      "BillAddress": "1240 road sinkholes"
      "BillZip": "06560"
      "BillCity": "Valbonne"
      "BillCountry": "France"
      «credits»: "1500"
    }
    {
      ...
    }
  ]
}

```

## VARIABLES RETURN

<b>getSubAccounts</b>	: Is the tag of the highest level of the stream.
<b>total</b>	: Total number of your sub accounts.
<b>subaccounts</b>	: Contains all your accounts under
<b>subaccount</b>	: Contains the details of a sub account see below.
<b>Login</b>	: Sub Account Name
<b>LastName</b>	: Name
<b>FirstName</b>	: First name
<b>ClientType</b>	: Customer type (company or individual)
<b>Company</b>	: Company name
<b>Phone</b>	: Phone
<b>Mobile</b>	: Mobile phone
<b>Fax</b>	: Fax
<b>E-mail</b>	: E-mail adress
<b>WebSite</b>	: Sub Account Website

<b>address</b>	: Address
<b>Zip</b>	: Postal code
<b>City</b>	: City
<b>Country</b>	: Country
<b>CreationDate</b>	: Creation date
<b>LastVisit</b>	: Last connection manager allmysms.com
<b>LastActivity</b>	: Last use of the API or site visit
<b>BillLastName</b>	: Billing Name
<b>BillFirstName</b>	: Billing Name
<b>BillCompany</b>	: Billing Company Name
<b>BillPhone</b>	: Billing Phone
<b>BillMobile</b>	: Mobile Billing
<b>BillEmail</b>	: Billing Email
<b>BillAddress</b>	: Billing address
<b>BillZip</b>	: Postal Code Billing
<b>BillCity</b>	: Billing City
<b>BillCountry</b>	: Billing Country
<b>Credits</b>	: Credits

#### OTHER VARIABLES RETURN

If you do not have sub account linked to your account, the return will only contain these tags:

**status** : 0  
**statusText** : No subaccount found

## 19. CREDITS MANAGEMENT FOR SUB ACCOUNT (MANAGESUBACCOUNTCREDITS)

### 19.1. OVERVIEW

Credit Management of a sub account (addition or deletion) directly from the API using the `manageSubAccountCredits` method.

#### ADDRESS

<https://api.allmysms.com/http/9.0/manageSubAccountCredits/>

#### SETTINGS

Parameter	Description	Mandatory
<b>login</b>	allmysms.com customer login	X
<b>apiKey</b>	API key	X
<b>subaccount</b>	sub account Login crediting	X

<b>credits</b>	Credits to add or remove sub account	X
<b>returnformat</b>	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 19.2. FEED BACK

The format of the response stream will match the one used when sending, XML or JSON.

### TREE

#### XML FORMAT

```
<ManageSubAccountCredits>
  <status> OK </ status>
  <subaccount>
    <Login>subaccountx </ Login>
    <credits> 1500 </ Credits>
  </subaccount>
</ ManageSubAccountCredits>
```

#### JSON FORMAT

```
{
  "Subaccount": [
    {
      "Login": "subaccountx"
      «credits": "1500"
    }
  ]
}
```

### VARIABLES RETURN

<b>managerSubAccountCredits</b>	: Is the tag of the highest level of the stream.
<b>status</b>	: "OK" if the sub account has been changed
<b>subaccount</b>	Contains the details of a sub account see below.
Login	: Sub Account Login
Credits	: Credits available

### OTHER VARIABLES RETURN

If the operation could not be done, the return will contain an error number and a corresponding statusText

Status	StatusText
0	The sub account is not linked to the master account
1	The "subaccount" parameter is not defined
2	The "credits" parameter is not defined
3	The account is under subscription
4	The sub account has insufficient funds (in the case of a debit)

5

The master account does not have enough credits (in the case of a credit)

## 20. REMOVING SUB ACCOUNT (DELETESUBACCOUNT)

### 20.1. OVERVIEW

To delete a sub-account, use the method deleteSubAccount.

#### ADDRESS

<https://api.allmysms.com/http/9.0/deleteSubAccount/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
loginToDelete	Login in the account to delete	X
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

### 20.2. FEED BACK

The format of the response stream will match the one used when sending, XML or JSON.

#### TREE

##### XML FORMAT

```
<DeleteSubAccount>
  <status> 1 </ status>
  < statusText> Sub-account has-been successfully deleted </ statusText>
</ DeleteSubAccount>
```

##### JSON FORMAT

```
{
  «status»: "1"
  «statusText»: "sub-account has-been successfully deleted"
}
```

#### VARIABLES RETURN

**status** : 1 if successful, 0 if an error  
**statusText** : Description of the processing of deleting the account under

## 21. CREATE CONTACT LIST (CREATELIST)



## 21.1. OVERVIEW

Use createList method to create contact lists. These lists will automatically appear in the customer area <https://manager.allmysms.com> of the account.

### ADDRESS

<https://api.allmysms.com/http/9.0/createList/>

### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
listData	stream containing the information to create your contact list	X

## 21.2. LISTDATA

### GENERAL STRUCTURE

XML or JSON feed must contain the corresponding different variables and values to your SMS.

XML feed example: see APPENDIX 11: Sample XML feed list creation

JSON feed example: see APPENDIX 12: Sample JSON feed list creation

### VARIABLES

List of variables sent in the XML stream / JSON (VARIABLE NAME: Definition)

- LISTNAME** (Required) Name of the relevant list.
- subaccount** : (Optional) Login under the account that created the list
- LOCKED** **(0 or 1)**, To lock the list so that it is not editable from your account
- STRUCTURE:**
- FIELDNAME** : **(mandatory)** Name of the field.
- FILTER** : **(0 or 1)**, This field acts as a filter when your sendings since allmysms manager.
- MAILING** **(0 or 1)**, direct mail field, you can customize the message when sending from the client area allmysms.com

## 21.3. FEED BACK

The format of the response stream will match the one used when sending, XML or JSON.

### TREE

#### XML FORMAT

```
<CreateList>
  <status> 1 </ status>
  <statusText> database successfully populated </ statusText>
</ CreateList>
```

### JSON FORMAT

```
{
  «status»: "1"
  «statusText" "database successfully populated"
}
```

### VARIABLES RETURN

**status**                Status of the request (0 in case of error or 1 if successful)

**statusText**        : Description of the insertion of one or more return contacts

## 22. INSERTING CONTACTS (POPULATELIST)

### 22.1. OVERVIEW

Insert one or more contacts using populateList method.

#### ADDRESS

<https://api.allmysms.com/http/9.0/populateList/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
populateData	stream containing the information or contacts to insert. The stream must contain the fields in the contact list previously created.	X

**ⓘ Warning:** If the stream contains nonexistent field, the contact will be ignored.

### 22.2. POPULATEDATA

#### GENERAL STRUCTURE

XML or JSON feed must contain the corresponding different variables and values to your SMS.

XML feed example:     see APPENDIX 13: Example of contact insertion XML feed

JSON feed example:   see APPENDIX 14: Sample JSON feed contacts insertion

#### VARIABLES

List of variables sent in the XML stream / JSON (VARIABLE NAME: Definition)

**LISTNAME** (Required) Name of the relevant list.  
**subaccount** : (Optional) Login under the account that created the list  
**CONTACTS:**  
**MOBILEPHONE: (mandatory)** contact phone number to insert.  
**FIELD NAME** : Value filled field

## 22.3. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```
<PopulateList>
  <status> 1 </ status>
  <statusText> list cavavin2 successfully populated </ statusText>
  <Success> 0 </ success>
  <Ignored> 1 </ ignored>
  <Duplicated> 0 </ duplicated>
</ PopulateList>
```

#### JSON FORMAT

```
{
  «status»: "1"
  «statusText" "list your base successfully populated"
  "Success": 1,
  "Ignored": 3
  "Duplicated": 2
}
```

### VARIABLES RETURN

**status** : status of the request (0 in case of error or 1 if successful) (cf.APPENDIX 15: Chart RETURNS contacts insertion)  
**statusText** : Description of the return  
**success**: Number of inserted contacts  
**ignored** : Number of contacts ignored (invalid number)  
**duplicated** : Number of double contacts found

## 23. REMOVAL CONTACTS (DELETECONTACTS)

### 23.1. OVERVIEW

Delete one or more contacts using the deleteContacts method.

#### ADDRESS

<https://api.allmysms.com/http/9.0/deleteContacts/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
deleteData	stream containing the information or contacts to remove. The stream must contain the fields in the contact list previously created.	X

## 23.2. DELETEDATA

### GENERAL STRUCTURE

XML or JSON feed must contain the corresponding different variables and values to your SMS.

XML feed example: see APPENDIX 16: Sample XML stream contacts suppression

JSON feed example: see APPENDIX 17: Sample JSON stream contacts suppression

### VARIABLES

List of variables sent in the XML stream / JSON (VARIABLE NAME: Definition)

**LISTNAME** (Required) Name of the relevant list.

**subaccount** : (Optional) Login under the account that created the list

**CONTACTS:**

**MOBILEPHONE:** (Required) Contact phone number to insert.

## 23.3. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```
<DeleteContacts>
  <status> 1 </ status>
  <statusText> list your base successfully updated </ statusText>
  <Success> 0 </ success>
  <Ignored> 6 </ ignored>
  <Duplicated> 0 </ duplicated>
</ DeleteContacts>
```

#### JSON FORMAT

```
{
  «status»: "1"
  «statusText" "list your base successfully updated"
  "Success": 0,
  "Ignored": 6
  "Duplicated": 0
}
```

## VARIABLES RETURN

<b>status</b>	: status of the request (0 in case of error or 1 if successful) (cf.ANEX 18: contacts suppression RETURNS Table)
<b>statusText</b>	: Description of the return
<b>success:</b>	: Number of deleted contacts
<b>ignored</b>	: Number of contacts ignored (invalid number)
<b>duplicated</b>	: Number of double contacts found

## 24. DELETING A CONTACT LIST (DELETEDLIST)

### 24.1. OVERVIEW

Delete a contact list using the deleteList method. This action is irreversible.

#### ADDRESS

<https://api.allmysms.com/http/9.0/deleteList/>

#### SETTINGS

Parameter	Description	Mandatory
<b>login</b>	allmysms.com customer login	X
<b>apiKey</b>	API key	X
<b>listName</b>	name of the contact list to delete	X
<b>subaccount</b>	Sub account login for which to perform the action	
<b>returnformat</b>	desired type of return XML or JSON (if not specified, the return will be JSON)	

### 24.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

#### TREE

##### XML FORMAT

```
<DeleteList>
  <status> 1 </ status>
  <statusText> list your base has-been successfully deleted </ statusText>
</ DeleteList>
```

##### JSON FORMAT

```
{
  «status»: "1"
  «statusText»: "your basic list has-been successfully deleted"
```

}

## VARIABLES RETURN

**status** : status of the request (0 in case of error or 1 if successful)  
**statusText** : Description of the return

## 25. RETRIEVING CONTACT LISTS (GETLISTS)

### 25.1. OVERVIEW

Use getLists method to retrieve the names of all the contact lists of account.

#### ADDRESS

<https://api.allmysms.com/http/9.0/getLists/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
subaccount	Sub account login for which to perform the action	
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

### 25.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

#### TREE

##### XML FORMAT

```

<GetLists>
  <Lists>
    <List>
      <ListName> base name </ listName>
      <Contact> 120 </ contact>
      <Locked> 1 </ locked>
      <CreationDate> 2015-01-25 </ creationDate>
    </ List>
    <List>
      <ListName> base name </ listName>
      <Contact> 2 </ contact>
      <Locked> 0 </ Locked>
      <CreationDate> 2015-02-05 </ creationDate>
    </ List>
  </ Lists>
</ GetLists>
    
```

## JSON FORMAT

```
{
  "Lists": [
    {
      "ListName" "base name"
      "Contacts", "120",
      "Locked": "1"
      "CreationDate": "2015-01-25"
    }
    {
      "ListName" "base name"
      "Contacts", "2",
      "Locked": "0"
      "CreationDate": "2015-02-05"
    }
  ]
}
```

## VARIABLES RETURN

**getLists:** Is the tag of the highest level of flow

**lists** : Contains the details of your contact lists

**listName** : name of the list

**contacts** : Number of contacts

**locked** 1 if the table is locked (read only), and 0 otherwise

**creationDate** : Creation date of the contact list

## 26. RETRIEVING CONTACTS FROM A LIST (GETCONTACTS)

### 26.1. OVERVIEW

Use getContacts method to retrieve the contacts stored in a list.

#### ADDRESS

<https://api.allmysms.com/http/9.0/getContacts/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
listName	name of the contact list to delete	X
subaccount	Sub account login for which to perform the action	
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

### 26.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

## TREE

### XML FORMAT

```
<GetContacts>
  <Contacts>
    <Contact>
      <MobilePhone> 336xxxxxxxx </ MobilePhone>
      <Firstname> Michel </ Firstname>
      <Lastname> Dupont </ Lastname>
      <Shop> allmysms </ Shop>
    </ Contact>
    <Contact>
      <MobilePhone> 336xxxxxxxx </ MobilePhone>
      <Firstname> Francois </ Firstname>
      <Lastname> Dupont </ Lastname>
      <Shop> allmysms </ Shop>
    </ Contact>
  </ Contacts>
</ GetContacts>
```

### JSON FORMAT

```
{
  "Contacts": [
    {
      "MobilePhone": "336xxxxxxxx"
      "Firstname": "Michel"
      "Lastname": "Dupont"
      "Shop": "allmysms"
    }
    {
      "MobilePhone": "336xxxxxxxx"
      "Firstname": "Francois"
      "Lastname": "Dupont"
      "Shop": "allmysms"
    }
  ]
}
```

## VARIABLES RETURN

- getContacts** : Is the tag of the highest level of flow
- contacts** : Contains details of returned contacts
- MobilePhone** : Contact phone number
- Column** : Contains the contact details

## 27. RETRIEVING THE NUMBER OF CREDITS PER ISO CODE (GETCREDITSBYCOUNTRYISOCODE)

### 27.1. OVERVIEW

Use `getCreditsByCountryIsoCode` method to retrieve the number of credits a SMS to a country through its ISO code.

## ADDRESS



<https://api.allmysms.com/http/9.0/getCreditsByCountryIsoCode/>

## SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
countryIsoCode	ISO code of the country in capitals (eg FR, DE, UK, ....)	X
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 27.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

## TREE

### XML FORMAT

```
<GetCreditsByCountryIsoCode>
  <IsoCode> EN </ IsoCode>
  <NbCredits> 15 </ nbCredits>
</ Getcreditsbycountrycode>
```

### JSON FORMAT

```
{
  "IsoCode ":" EN "
  "nbCredits ":" 15 "
}
```

## VARIABLES RETURN

**getCreditsByCountryIsoCode** : Is the tag of the highest level of flow  
**IsoCode** : contains ISO code of the desired country  
**nbCredits** : number of credits for an SMS sent to that country

## 28. CREATING A SHORTENED URL (SHORTENURL)

### 28.1. OVERVIEW

Use shortenUrl method to generate a shortened URL of a given URL.

## ADDRESS

<https://api.allmysms.com/http/9.0/shortenUrl/>

## SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
url	Url shortening, must be accessible.	X
returnformat	desired type of return XML or JSON (if not specified, the return will be JSON)	

## 28.2. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

### TREE

#### XML FORMAT

```
<ShortenUrl>
  <Url> http://www.yoururl.fr </ url>
  <SHORTURL> http://bs.ms/xxxx </ SHORTURL>
</ ShortenUrl>
```

#### JSON FORMAT

```
{
  "url ":" http: \ / \ / www.yoururl.fr "
  "SHORTURL ":" http: \ / \ / bs.ms \ / xxxx "
}
```

### VARIABLES RETURN

**shortenUrl** : Is the tag of the highest level of flow  
**url** contains Url shortening  
**SHORTURL** Contains the shortened URL

## 29. SENDING EMAIL (SENDEMAIL)

### 29.1. OVERVIEW

Use the sendEmail API method to achieve your outgoing emails.

#### ADDRESS

<https://api.allmysms.com/http/9.0/sendEmail/>

#### SETTINGS

Parameter	Description	Mandatory
login	allmysms.com customer login	X
apiKey	API key	X
emailData	XML or JSON feed containing the mail content and targeted email addresses	X

## 29.2. EMAILDATA

### GENERAL STRUCTURE

XML or JSON feed must contain the corresponding different variables and values to your outgoing emails.

XML feed example: see ANNEX 19: Example of sending XML stream EMAIL

JSON feed example: see ANNEX 20: Example of sending JSON stream EMAIL

### VARIABLES

List of variables sent in the XML stream / JSON (VARIABLE NAME: Definition)

**FROM:**Email (valid) sender

**TO:**targeted email addresses

**SUBJECT:**Email Subject

**DATE:**[Optional] date in the YYYY-MM-DD HH: MM: SS in case of delayed transmission programming

**HTML:**HTML source code of the email base64 encoded

**ALIAS:**[Optional] The name that will be displayed instead of the email sender

**MAILTEXT:** [Optional] Contents of the email text version (for browsers not accepting HTML content)

**CAMPAIGN\_NAME:** [Optional] specifies a campaign name for later identification if necessary

**REPLY\_TO:**[Optional] reply email address

## 29.3. FEED BACK

The feedback format will be the same as the one used for sending (either XML or JSON).

Each API call will return the status of your request (Status), his narrative explanation (StatusText), your remaining credit (Credits).

If an error occurs during the HTTP request, a copy of the error will be sent by email (if requested). If your problems persist,contact support.

### STRUCTURE OF THE RESPONSE

#### 29.3.1.1. EXAMPLE

##### XML FORMAT

```
<SendSMS>
  <status> 100 </ status>
```

```
<statusText> Your email-have-been feeling </ statusText>
<campaignId> xxxxxxxxxx </ campaignId>
<InvalidEmails> xxxxxxxxxx </ invalidEmails>
<credits> 370 </ credits>
</ SendSMS>
```

## JSON FORMAT

```
{
  «status»: 100,
  «statusText»: "Your email-have-been feeling"
  "campaignId": "xxxxxxxxxxxx"
  «invalidEmails»: "xxxxxxxxxxxx"
  «credits", "340"
}
```

### 29.3.1.2. VARIABLES RETURN

<b>status:</b>	Statutes of the request (status list: cf.21 ANNEX: Table of Statutes after sending)
<b>statusText:</b>	textual explanation
<b>credits:</b>	number of remaining credits
<b>campaignId:</b>	identification of the email campaign conducted

## 30. SMS BACKOFFICE

You can view your campaigns, fund your account, create sub-accounts, send SMS from the web, or view your statistics and invoices on your secure client area at the following address:

<https://manager.allmysms.com>



Customer area in responsive design, displaying properly on your computer as well as on tablet or smartphone.

## 31. GLOSSARY

**Acknowledgment:**

Acquittal on a shipment. Allows for example to know the status of the receipt of a message

**DLR (Delivery Report):**

Delivery Report provided by mobile operators to know if a message has been sent to recipient.

**HTTP (HyperText Transfer Protocol):**

Internet protocol used to transfer data between a client application and a web server.

**JSON (JavaScript Object Notation)**

JSON is a textual data format, generic, derivative notation objects ECMAScript language. He used to represent structured information.

**IASP (Do More lives at this address):**

Refers to a number that is not assigned (indicating by an Operator DLR).

**SMS (Short Message Service)**

Designates the messaging service for sending and receiving alphanumeric or binary messages to or from a mobile device.

**SMS-C (SMS Center):**

Mobile Operator host that manages the sending and receiving of SMS-MO and SMS-MT.

**MO-SMS (Short Message Service Mobile Originated)**

Denotes a SMS sent by a user from his mobile terminal.

**SMS MT (Mobile Terminated Short Message Service)**

Denotes an SMS received by user on his mobile terminal.

**SMS Push Pull:**

Literally, Push-Pull. This is actually a two-way model of sending SMS-MT (Push) to solicit return SMS-MO responses (Pull).

So you send your SMS MT campaign to all your contacts directly and receive their answers by SMS-MO.

**STOP:**

All SMS sending commercial nature must be associated with the ability to unsubscribe from the service. For this, send in response STOP keyword is automatically taken into account. This is possible directly to the originating number if sent without customization of the issuer. When customizing, the message must contain a statement of the type "STOP to 36XXX." The user has the possibility to re-subscribe by sending START or REABO the same number.

**XML (eXtensible Markup Language)**

Evolution of SGML, XML allows designers to HTML documents to define their own tags in order to customize the data structure they intend to present. So that HTML specifies how the elements of a page will be presented, XML defines what those elements contain.

## 32. NOTES

### 32.1. APPENDIX 1: SAMPLE XML SMS SENDING STREAMS

```

<DATA>
<MESSAGE> <! [CDATA [Your message]]> </ MESSAGE>
<DYNAMIC>2 (will contain the number of variable message parameters) </ DYNAMIC> (required if the message is
variable)
<CAMPAIGN_NAME>Campaign          Name          </          CAMPAIGN_NAME>          (optional          extra)
<DATE>optional deferred ex .: Date 2011-11-05 3:10:00 p.m. </ DATE> (optional extra)
<TPOA>station name (optional extra) </ TPOA>
<MAIL NOTIF>0 -> no sending email notifications, 1 -> Notification </ MAIL_NOTIF> (optional extra)
<Text>
    <MOBILEPHONE> 336xxxxxxxx </ MOBILEPHONE>
    <PARAM_1>Parameter 1 </ PARAM_1>
    <PARAM_2>Parameter 2 </ PARAM_2>
</ Text>
<Text>
    <MOBILEPHONE> 3361111111 </ MOBILEPHONE>
    <PARAM_1>Parameter 1 </ PARAM_1>
    <PARAM_2>Parameter 2 </ PARAM_2>
</ Text>
</ DATA>
    
```

### 32.2. APPENDIX 2: EXAMPLE OF SMS SENDING JSON FEED

```

{
  "DATA": {
    "CAMPAIGN_NAME": "Campaign Name"
    "MESSAGE": "Your message"
    "TPOA": "Name of the issuer"
    "DYNAMIC": "2"
    "DATE": "2013-01-01 12:00:00"
    "SMS": [
      {
        "MOBILEPHONE": "336xxxxxxxx"
        "PARAM_1": "parameter 1"
        "PARAM_2": "parameter 2"
      }
      {
        "MOBILEPHONE": "3361111111"
        "PARAM_1": "parameter 1"
        "PARAM_2": "parameter 2"
      }
    ]
  }
}
    
```

### 32.3. APPENDIX 5: TABLE OF STATUTES AFTER SENDING

Status	StatusText
100	The message has been sent
101	The message was scheduled to be sent at
102	Connection Problem - No account matches the specified clientcode and apiKey

104	insufficient credit to handle the shipment. Use: XX Credits, Available: YY Credits. Please re-credit your account on AllMySMS.com
105	Empty XML Feeds
109	invalid or incomplete XML after the <MESSAGE>
110	undefined message (empty) in the XML stream
111	The message exceeds the number of characters
114	No valid phone number in the stream. Please refer to the online documentation to know the valid formats.
115	Invalid XML feed or badly formatted dates between the <DATE> and </ DATE>

#### 32.4. ANNEX 6: TABLE OF STATUTES OF ACCUSED

Status	StatusText
1	issued
2	Not issued (sent by the operator)
3	Transmitted to the operator
4	Message rejected
5	SMS rejected (probably unknown number / missing subscriber)

#### 32.5. APPENDIX 7: COUNTRY SUPPRESSION STATUS TABLE

Status	StatusText
CAMPIDNOTFOUND	Identification Problem
CAMPIDNOTFOUND	Campid parameter missing
CAMPNOTFOUND	The campaign was not found
BADCAMPAIGNSENDINGTYPE	The campaign is not scheduled



<b>CAMPAIGNALREADYDELETED</b>	Campaign already deleted
<b>BCAMPAIGNSTATUS</b>	The status of the campaign is wrong
<b>BADDATETOSEND</b>	The mailing date is corrupted
<b>CAMPALREADYSENT</b>	The campaign has already been sent
<b>OK</b>	The campaign has been deleted

## 32.6. APPENDIX 8: SAMPLE XML STREAM IN CREATION ACCOUNT

```

<DATA>
<FIRSTNAME> Thomas </ FIRSTNAME>
<LASTNAME> SMITH </ LASTNAME>
<SOCIETY> Society </ SOCIETY>
<MOBILE> + 336XXXXXXXX </ MOBILE>
<EMAIL> email@domain.com </ EMAIL>
<LOGIN> newlogin </ LOGIN>
<PASSWORD> newpass </ PASSWORD>
<PHONE> 0493000000 </ PHONE>
<WEBSITE> domain.com </ WEBSITE>
<ADDRESS> address new account </ ADDRESS>
<ZIP> 01xxxx </ ZIP>
<CITY> Paris </ CITY>
<COUNTRY> FRANCE </ COUNTRY>
<SENDEMAILTOCUSTOMER> 1 </ SENDEMAILTOCUSTOMER>
<SENDEMAILTOMASTERACCOUNT> 1 </ SENDEMAILTOMASTERACCOUNT>
<POSTPAID> 1 </ POSTPAID>
<PROFILE> to define MS Innovations </ PROFILE>
</ DATA>
    
```

## 32.7. APPENDIX 9: EXAMPLE OF JSON STREAM UNDER CREATION ACCOUNT

```

{
  "DATA": {
    "FIRSTNAME": "Thomas"
    "LASTNAME": "Dupont"
    "SOCIETY": "Society"
    "MOBILE": "+ 336XXXXXXXX"
    "EMAIL": "email@domain.com"
    "LOGIN": "newlogin"
    "PASSWORD": "newpass"
    "PHONE": "0493000000"
    "WEBSITE": "domain.com"
    "ADDRESS": "address new account",
    "ZIP": "01xxxx"
    "CITY": "Paris"
    "COUNTRY": "FRANCE"
    "SENDEMAILTOCUSTOMER": "1"
    "SENDEMAILTOMASTERACCOUNT": "1"
    "POSTPAID": "1"
    "PROFILE": "to define MS Innovations"
  }
}
    
```

## 32.8. 10 ANNEX: TABLE OF CREATING RETURNS / MODIFICATION SUB ACCOUNT

Status	StatusText
0	An error has occurred
1	Creating / Changing done subaccount

## 32.9. APPENDIX 11: SAMPLE XML FEED LIST CREATION

```
<DATA>
  <LISTNAME>mylist </ LISTNAME>
  <LOCKED> 0 </ LOCKED>
  <STRUCTURE>
    <FIELD>
      <FIELDNAME> City </ FIELDNAME>
      <FILTER> 1 </ FILTER>
      <MAILING> 0 </ MAILING>
    </ FIELD>
    <FIELD>
      <FIELDNAME>Surname </ FIELDNAME>
      <FILTER> 0 </ FILTER>
      <MAILING> 1 </ MAILING>
    </ FIELD>
  </ STRUCTURE>
</ DATA>
```

## 32.10. APPENDIX 12: SAMPLE JSON FEED LIST CREATION

```
{
  "DATA": {
    "LISTNAME": "mylist"
    "LOCKED": "0"
    "Structure": [
      {
        "FIELDNAME": "City"
        "FILTER": "1"
        "MAILING": "0"
      }
      {
        "FIELDNAME": "Surname"
        "FILTER": "0"
        "MAILING": "1"
      }
    ]
  }
}
```

## 32.11. APPENDIX 13: EXAMPLE OF CONTACT INSERTION XML FEED

```
<DATA>
  <LISTNAME> Your Base </ LISTNAME>
  <CONTACTS>
    <CONTACT>
      <FIRSTNAME>xxxx </ FIRSTNAME>
      <LASTNAME>xxxx </ LASTNAME>
      <MOBILEPHONE> 06xxxxxx </ MOBILEPHONE>
      <Store>allmysms </ SHOP>
    </ CONTACT>
    <CONTACT>
      <FIRSTNAME>yyyy </ FIRSTNAME>
      <LASTNAME>yyyy </ LASTNAME>
      <MOBILEPHONE> 06xxxxxx </ MOBILEPHONE>
      <Store>allmysms </ SHOP>
    </ CONTACT>
  </ CONTACTS>
</ DATA>
```

## 32.12. APPENDIX 14: SAMPLE JSON FEED CONTACTS INSERTION

```
{
  "DATA": {
    "LISTNAME": "Your Base"
    "CONTACT": [
      {
        "FIRSTNAME": "xxxx"
        "LASTNAME": "xxxx"
        "MOBILEPHONE": "06xxxxxx"
        "SHOP": "allmysms"
      }
      {
        "FIRSTNAME": "yyyy"
        "LASTNAME": "yyyy"
        "MOBILEPHONE": "06xxxxxx"
        "SHOP": "allmysms"
      }
    ]
  }
}
```

## 32.13. APPENDIX 15: CHART RETURNS CONTACTS INSERTION

Status	StatusText
1	The contact has been added
0	The contact could not be added

## 32.14. APPENDIX 16: SAMPLE XML STREAM CONTACTS SUPPRESSION

```
<DATA>
<LISTNAME> cavavin2 </ LISTNAME>
<CONTACTS>
<CONTACT>
<MOBILEPHONE> 0600000000 </ MOBILEPHONE>
</ CONTACT>
<CONTACT>
<MOBILEPHONE> 0611111111 </ MOBILEPHONE>
</ CONTACT>
</ CONTACT>
</ DATA>
```

## 32.15. APPENDIX 17: SAMPLE JSON STREAM CONTACTS SUPPRESSION

```
{
  "DATA": {
    "LISTNAME": "your basic"
    "CONTACT": [
      {
        "MOBILEPHONE": "0600000000"
      }
      { "MOBILEPHONE": "0611111111" }
    ]
  }
}
```

## 32.16. ANNEX 18: CONTACTS SUPPRESSION RETURNS TABLE

Status	StatusText
1	The contact has been deleted
0	The contact could not be deleted

### 32.17. ANNEX 19: EXAMPLE OF SENDING XML STREAM EMAIL

```

<DATA>
<FROM> email address of the sender </ FROM>
<TO> email addresses of the recipients separated by </ TO>
<SUBJECT> Object of the email </ SUBJECT>
<DATE>Delayed Date ex .: 2016-02-05 3:10:00 p.m. </ DATE> (optional extra)
<HTML> email HTML source code of base64 encoded </ HTML>
<ALIAS> Name displayed instead of the sender email address </ ALIAS> (optional extra)
<MAILTEXT> Content of the email text Version </ MAILTEXT> (optional extra)
<CAMPAIGN_NAME> Campaign Name </ CAMPAIGN_NAME> (optional extra)
<REPLY_TO>reply email address </ REPLY_TO> (optional extra)
</ DATA>
    
```

### 32.18. ANNEX 20: EXAMPLE OF SENDING JSON STREAM EMAIL

```

{
  "DATA": {
    "FROM" "valid email address of the sender"
    "TO", "targeted valid email addresses"
    "SUBJECT": "Email Subject"
    "DATE": "(optional) 2016-01-01 12:00:00"
    "HTML", "HTML source code of the email base64 encoded"
    "ALIAS": "(optional) Name that will appear instead of the sender email"
    "MAILTEXT": "(optional) Content of the email text version of"
    "CAMPAIGN_NAME": "(Optional) Name of the campaign"
    "REPLY_TO": "(optional) Email address response"
  }
}
    
```

### 32.19.21 ANNEX: TABLE OF STATUTES AFTER SENDING

Status	StatusText
100	The message has been sent
101	The message was scheduled to be sent at
102	Connection Problem - No account matches the specified clientcode and apiKey
104	insufficient credit to handle the shipment. Use: XX Credits, Available: YY Credits. Please re-credit your account on AllMySMS.com
105	Empty XML / JSON
106	Email address of the sender invalid or void

107	Purpose of the blank email
109	Invalid XML / JSON or incomplete after the <HTML>
110	Emails undefined recipients (empty) in the XML stream
114	No valid email phone in the stream. Please refer to the online documentation to know the valid formats.
115	Invalid XML feed or badly formatted dates between the <DATE> and </ DATE>

### 32.20.24 ANNEX: TABLE OF STATUTES AFTER SENDING

Status	StatusText
100	The message has been sent
101	The message was scheduled to be sent at
102	Connection Problem - No account matches the specified clientcode and apiKey
104	insufficient credit to handle the shipment. Use: XX Credits, Available: YY Credits. Please re-credit your account on AllMySMS.com
105	XML / JSON is empty
110	URL empty voice file
111	Date of dispatch badly formatted
112	Invalid Shipper Number
113	No valid phone number
114	Invalid voice file. Accepted Format: WAV or MP3
115	Size incorrect voice file (must be between 1K and 5MB)
116	Time invalid voice file (must be between 5 and 35 seconds)