



# API SMS

## Revisions

Revision	Author	Date	Comments
1	pmarechal	21/06/2016	English version
2	vroche	13/04/2017	REST API
3	pmarechal	03/08/2017	Update the SMTP section
4	pmarechal	15/09/2017	Description of the smsstatus resource
5	pmarechal	26/09/2017	How to retrieve responses (SMS MO)
6	pmarechal	21/11/2017	How to customise the sender id
7	pmarechal	29/12/2017	How to specify that we want receive SMS responses
8	athomas	10/11/2020	Specify that ping requests are done with HEAD http verb

# Table des matières

## [1. Overview](#)

### [1.1. Authentication](#)

[1.1.1. Retrieving the application ids](#)

[1.1.2. Retrieve an authentication token](#)

[1.1.3. Using the authorization token \(access token\)](#)

### [1.2. Pagination, sorting and filtering](#)

#### [1.2.1. Pagination](#)

[1.2.1.1. introduction](#)

[1.2.1.2. Example 1: Limiting results](#)

[1.2.1.3. Example 2: Pagination](#)

#### [1.2.2. Sorting](#)

[1.2.2.1. Introduction](#)

[1.2.2.2. Example](#)

#### [1.2.3. Limiting the list of attributes returned per resource](#)

[1.2.3.1. Introduction](#)

[1.2.3.2. Example 1](#)

[1.2.3.3. Example 2](#)

### [1.3. The lists of resources returned](#)

### [1.4. The various actions on a resource](#)

[1.4.1. Introduction](#)

[1.4.2. Example](#)

[1.4.3. Methods profile for each action](#)

### [1.5. Updating resources](#)

### [1.6. Return codes](#)

### [1.7. Response formats](#)

[1.7.1. Introduction](#)

[1.7.2. Examples](#)

[1.7.3. Cross-domain](#)

### [1.8. Filters and passing multiple values](#)

### [1.9. Integrating our API as PHP](#)

## [2. API possibilities](#)

[2.1. Text message](#)

[2.2. Long text message](#)

[2.3. Binary text message](#)

[2.4. International text messages](#)

[2.5. Mobile numbers](#)

[2.6. Scheduled text messages](#)

[2.7. Volume](#)

[2.8. Text message statuses](#)

[2.9. Limitations](#)

[2.10. Responses](#)

[2.11. Ping](#)

[2.12. Optional parameters](#)

### [3. HTTPS API](#)

[3.1. HTTPS address](#)

[3.2. Description of the sms resource](#)

[3.3. List of properties of the sms resource](#)

[3.4. Sending a simple text message](#)

[3.4.1. List of parameters to supply in order to send a simple text message](#)

[3.4.2. Example](#)

[3.5. Sending a personalized text message](#)

[3.5.1. List of parameters to supply in order to send a personalized text message](#)

[3.5.2. Example](#)

[3.6. Reading a text message](#)

[3.6.1. List of available filters](#)

[3.6.2. Attributes for a text message returned by default](#)

[3.6.3. Example](#)

[3.6.3.1. Example 1: Retrieving all the message send to one phone number](#)

[3.6.3.2. Example 2: Retrieving the message sent to one phone number at a specified date](#)

[3.6.3.3. Example 3: Retrieving all messages sent between 2 dates](#)

[3.6.3.4. Example 3: Retrieving all responses \(SMS MO: from mobile to Digitaleo\) of a mailing](#)

[3.7. Canceling a text message](#)

[3.7.1. List of available filters](#)

[3.8. Description of the mailing resource](#)

[3.8.1. List of properties of the mailing resource](#)

[3.8.2. List of parameters to supply in order to send a text message](#)

[3.8.3. Sending a simple text message mailing](#)

[3.8.4. Sending a personalized text message mailing](#)

[3.9. Reading a mailing](#)

[3.9.1. List of available filters](#)

[3.9.2. Example](#)

[3.10. Canceling a mailing](#)

[3.10.1. List of available filters](#)

[3.11. Retrieving the current status of text message](#)

[3.11.1. List of properties of the status resource](#)

[3.11.2. Reading the status of one or several message](#)

[3.11.2.1. List of available filters](#)

[3.11.2.2. Example 1: Read the current status of all messages from a mailing with its id.](#)

[3.11.2.3. Example 2: Read the current status of a message.](#)

[3.11.3. Retrieving a queue of status](#)

[3.11.3.1. List of available filters](#)

[3.11.3.2. Example 1: Retrieving all the status from the queue.](#)

[3.11.3.3. Example 2: Retrieving the 4 last status from the queue.](#)

[3.12. Reading all the status of one text message](#)

[3.12.1. List of properties of the smsstatus resource](#)

[3.12.2. List of available filters](#)

[3.12.3. Example](#)

[3.13. Description of the stats resource](#)

[3.13.1. List of properties of the statistic resource](#)

[3.13.2. Reading statistics](#)

[3.13.2.1. List of available filters](#)

[3.13.2.2. Example 1: Retrieving the statistics using a mailing name](#)

[3.13.2.3. Example 2: Retrieving the statistics from a month](#)

[4. SMTP API](#)

[4.1. E-mail address](#)

[4.2. Sending a text message mailing](#)

[4.3. Sending a customized text message mailing \(advanced mode\)](#)

[4.4. PHP Example](#)

[5. FTP API](#)

[5.1. FTP address](#)

[5.2. Sending a text message mailing](#)

[6. Texteo monitor website](#)

[6.1. Web address](#)

[6.2. Format](#)

[A. APPENDIX A - TEXT MESSAGE CHARACTERS](#)

[Copyright](#)

This document describes the interfaces, or APIs, used by Digitaleo's ecoSMS platform that you can use to send text messages, check messages status, and read replies.

These are the various interfaces:

- **HTTPS API: secure web** interface to send a small number of text messages – quick to implement
- **SMTP API: e-mail** interface (mail2sms) to send high volumes of mailings
- **FTP API: file transfer interface**, same use as SMTP API

Each interface can provide all or some of the following services:

- **sending a text message:** to send a text message to a recipient
- **receiving a text reply:** to get the replies sent by your text message recipients
- **sending a simple text message mailing:** to send the same text message to a high number of recipients
- **sending a personalized text message mailing:** to send a personalized text message to a high number of recipients
- **text message status check:** to monitor the status of one or more text messages over time using its ID, the ID or name of a mailing, or the most recent messages
- **canceling a text message or mailing:** to cancel a scheduled text message or mailing before it is sent (or even while it is being sent, in the case of mailings)
- **number checking:** checks mobile phone numbers before sending
- **text message statistics:** to see at a glance how many text messages you send by month, by status

The table below presents the availability of services and options by API:

	Services							Options			
	Sending a text message	Simple text mailing	Personalized text mailing	Text message status	Replies	Text message statistics	Number checking	Canceling text messages or mailings	Binary text message	Long text message	Scheduled text message
HTTPS API	X	X	X	X	X	X	X	X	X	X	X
SMTP API	X	X	X							X	
FTP API	X	X	X							X	

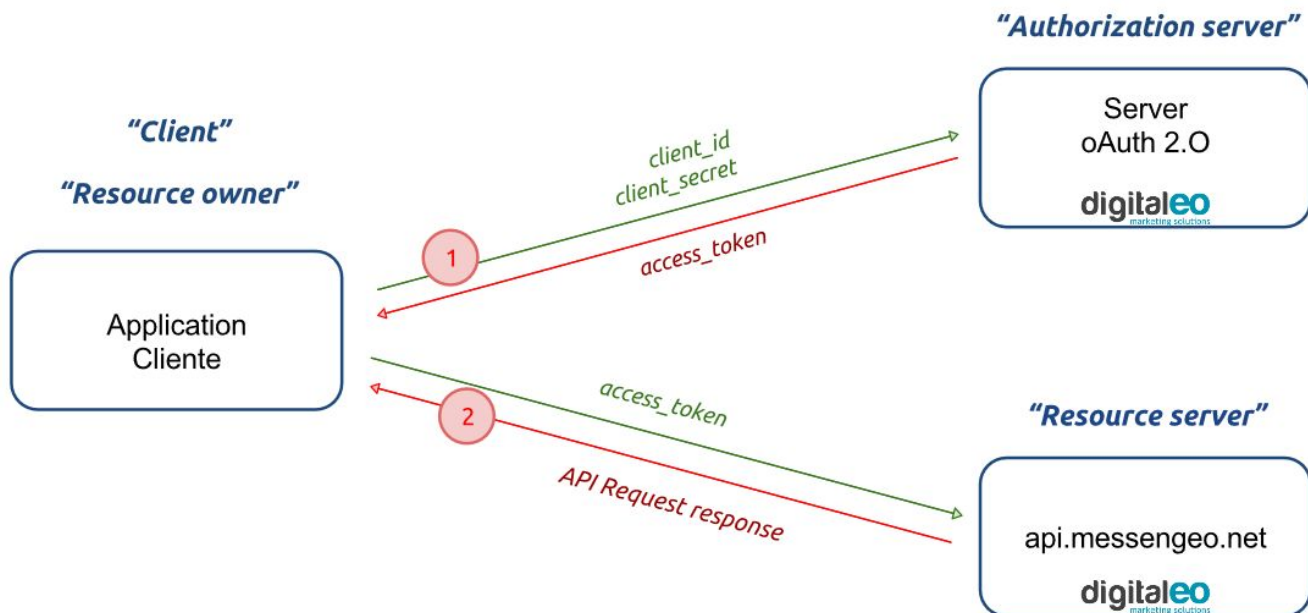
# 1. Overview

This API is not RESTfull because for all of the calls, the verbs HTTP GET and POST can be used. However, it is based on various resources of which the details are provided further on in this document.

The purpose of this first section is to help you understand the various types of calls to our APIs, regardless of the resource.

## 1.1. Authentication

Authentication to our APIs is based on the oAuth 2.0 protocol. Each call to our APIs has to contain an `access_token` that the client application will have requested beforehand from the Digitaleo authorization server:



### 1.1.1. Retrieving the application ids

To retrieve a `client_id` and a `client_secret`, you must declare an application in the Digitaleo platform.

For this,

1. Connect to [app.digitaleo.com](https://app.digitaleo.com)
2. Click on the Parameters menu
3. Go to the API tab

### 1.1.2. Retrieve an authentication token

The client must perform a POST request with the following parameters:

- `grant_type`: The value must be "client\_credentials" for this type of authorization
- `client_id`: The id of the application (client)
- `client_secret`: The secret key of the application (client)

*Note: The `client_id` and `client_secret` will be sent to you.*

The URL for retrieving a token is as follows

```
https://oauth.messengerge.net/token
```

Example of an HTTP request

```
POST /token HTTP/1.1
Host: oauth.messengerge.net
Content-Type: application/x-www-form-urlencoded

client_id=51612c780b4dbaea8f81995becbcbfec08969d0e&
client_secret=p280edbd76d510c41990cbe5e6108c7e&
grant_type=client_credentials
```

Example of a request with Curl

```
curl https://oauth.messengerge.net/token
-d 'client_id=51612c780b4dbaea8f81995becbcbfec08969d0e'
-d 'client_secret=p280edbd76d510c41990cbe5e6108c7e'
-d 'grant_type=client_credentials'
```



## Return

if successful, the authorization server will return a code 200 HTTP response of which the body will contain the following JSON flow

```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...",
  "expires_in": "3600",
  "token_type": "bearer",
  "scope": "basic",
}
```

Description of the various fields:

Property	Description
access_token	The token issued by the authorization server. <i>Note: The size of the token can range up to 50,000 characters</i>
expires_in	The lifespan in seconds of the token issued
token_type	The type of token. The Digitaleo server only supports the "bearer" type
scope	The scope of the token

If one of the parameters is not correct, the authorization server will return a code 400 http response (HTTP/1.1 400 Bad Request) of which the body will contain the following json flow:

```
{
  "error": "invalid_client",
  "error_description": "The client credentials are invalid",
}
```

### 1.1.3. Using the authorization token (access\_token)

The authorization token is sent to the API in the header of the HTTP request and more particularly in the header "Authorization: Bearer". Note that the "Authorization: Bearer" is case-sensitive.

Example of an HTTP request

```
GET /rest/campaigns HTTP/1.1

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
```

Example of a request with Curl

```
curl -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2..."
https://api.messengeo.net/rest/campaigns
```

## 1.2. Pagination, sorting and filtering

### 1.2.1. Pagination

#### 1.2.1.1. introduction

Three parameters allow you to manage the paginated resource display.

The parameters are

- **limit:** allows you to limit the number of elements returned
- **offset:** allows you to ignore the first n elements of the list
- **total:** allows you to retrieve the total number of resources if indeed the request had not limited the number of resources returned. It is therefore useful in the framework of using a limit for pagination. (the default is false). This feature uses a lot of resources. It is recommended that it not be activated in the case there is no pagination.

#### 1.2.1.2. Example 1: Limiting results

Retrieving only 20 resources and the total number of resources if the result were limited to 20 resources

```
GET /rest/resource HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

limit=20&total=true
```

with Curl

```
curl
-X GET
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG...ccOqbVow8xOQyQ"
-d limit=20
-d total=true
https://api.messengeo.net/rest/ressource
```

### 1.2.1.3. Example 2: Pagination

Retrieving 10 resources, leaving aside the first 20. This boils down to reading the 3rd page knowing that each page lists 10 resources.

```
GET /rest/resource HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

limit=10&offset=20
```

## 1.2.2. Sorting

### 1.2.2.1. Introduction

The **sort** parameter allows you to order the list of resources returned according to one of the attributes of the resource concerned. This parameter is comprised of two elements separated by a space:

- The name of the attribute according to which you want to order the list
- The sorting order, ascending order (ASC) or descending order (DESC)

### 1.2.2.2. Example

Sorting a list of resources in descending order according to the id of this resource:

```
GET /rest/resource HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

sort=id%20DESC
```

with Curl

```
curl
-X GET
-H "Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbG...ccOqbVow8xOQyQ"
-d sort=id DESC
https://api.messengeo.net/rest/ressource
```

## 1.2.3. Limiting the list of attributes returned per resource

### 1.2.3.1. Introduction

It is possible to limit the list of attributes of the resources returned. In other words, this entails returning incomplete resources in order to focus on the attributes that are really necessary for the client that generated the call.

This makes it possible to save both bandwidth and processing on the server side. Indeed, certain attributes are calculated at the time of the call and not retrieving them makes it possible to reduce the request time.

The parameter that allows you to define the attributed return is called `properties`.

For each resource, a list of attributes returned by default (if the `properties` parameter is not defined) is imposed. An alias called `DEFAULT` makes it possible to specify that you want to retrieve the attributes by default + another or – another.

### 1.2.3.2. Example 1

Retrieving only the id and the name of each resource

```
GET /rest/resource HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

properties=id,name
```

with Curl

```
curl
-X GET
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG...ccOqbVow8xOQyQ"
-d properties=id,name
https://api.messengeo.net/rest/ressource
```

### 1.2.3.3. Example 2

Retrieving the attributes returned by default except the id

```
GET /rest/resource HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

properties=DEFAULT,-id
```

with Curl

```
curl
-X GET
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG...cc0qbVow8xOQyQ"
-d properties=DEFAULT,-id
https://api.messengeo.net/rest/ressource
```

## 1.3. The lists of resources returned

Most of the calls to the REST API return a list of resources. This list of resources is comprised of the following three elements:

- **size:** The number of resources returned
- **total:** The number of resources returned if the request had not limited the result
- **list:** The table containing the resources

An example of a list of resources returned in json format

```
{
  "size": 2, // The number of resources returned
  "total": 600640, // The number of resources returned if the request had not limited the result
  "list": // The table containing the resources
  [
    {
      "id": "1",
      "email": "aladdin@digitaleo.com",
      "phone": "+33201010101",
      "mobile": "+33601010101",
    },
    {
      "id": "2",
      "email": "jasmine@digitaleo.com",
      "phone": "+33202020202",
      "mobile": "+33602020202",
    }
  ]
}
```

## 1.4. The various actions on a resource

### 1.4.1. Introduction

Our APIs comply with the HTTP verbs and their correspondence with the CRUD actions (Create, Read, Update, Delete) of a resource. However, it is also possible to perform all of the actions only with HTTP GET requests or only with HTTP POST requests. To do this, the action to be performed must be specified in the URL.

Description of the action	Dedicated HTTP verb	Name of the action
Read resources	GET	read
Create a resource	POST	create
Update resources	PUT	update
Delete resources	DELETE	delete

### 1.4.2. Example

The two following Curl requests are considered to be equivalent by our APIs

```
curl
-X GET
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG...ccOqbVow8xOQyQ"
https://api.messengeo.net/rest/ressource
```

```
curl
-X POST
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG...ccOqbVow8xOQyQ"
https://api.messengeo.net/rest/ressource?action=read
```

### 1.4.3. Methods profile for each action

Description of the action	input	output
read	Filter	List of resources
create	Parameters	Resource created
update	Filter + Parameter	Number of resources updated
delete	Filter	Number of resources deleted



## 1.5. Updating resources

Updating resources is based on two types of parameters:

- A list of filters that allows you to limit the resources to be updated
- A list of keys/values that allow you to define which resource attributes to update, with which value.

The keys/values to be updated must be grouped together in a "metaData" attribute, in the form of a JSON flow.

For example, the following request makes it possible to update the name and the description of resources for which the id is either 100, or 200 and for which the name is equal to "former name of the resource"

```
PUT /rest/resource HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

id=100,200&name='ancien nom de la ressource'&metaData={"name":"New name of the resource","description":"New
description of the resource"}
```

with Curl

```
curl
-X PUT
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...ccOqbVow8xOQyQ"
-d id=100,200
-d name='ancien nom de la ressource'
-d metaData={"name":"Le nouveau nom de la ressource","description":"La nouvelle description de la ressource"}
https://api.messengeo.net/rest/ressource
```

Actions that update resources return the number of contacts involved by the filter passed as input

```
{
  count: 10,
}
```

## 1.6. Return codes

The HTTP response code is contained:

- in the HTTP header,
- in the content of the response in the case of an error.

The return codes are based on the HTTP return codes:

- 2XX - The call to the API unfolded correctly
- 4XX - The call to the API has an error in its parameters.

Codes with success:

- **200 OK:** everything went well
- **201 Created:** Resource created
- **204 No Content:** Resource updated or deleted

The error codes that you are likely to see are the following:

- **304 Not Modified:** Error during updating or deleting (the resource is not modified)
- **400 Bad Request:** Missing or incorrect parameter
- **401 Unauthorized:** Authentication failed
- **403 Forbidden:** Access to the requested location is prohibited
- **404 Not Found:** Unknown method or method not indicated
- **405 Method Not Allowed:** You are not authorized to use the method that you are requesting
- **414 Request-URI Too Long:** Your request is too large, please shorten it
- **417 Expectation Failed:** The required parameters are either missing or are incorrect
- **500 Internal Server Error:** Unidentified error

For example, if the authentication token is no longer valid for the following request:

```
curl
-X GET
-H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbG...ccOqbVow8xOQyQ"
https://api.messengeo.net/rest/resource
```

The header of the HTTP response will be

```
< HTTP/1.1 401 Unauthorized
< Date: Fri, 06 Mar 2015 21:32:06 GMT
< Server: Apache/2.2.16 (Debian)
< X-Powered-By: PHP/5.3.3-7+squeeze15
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< Content-Length: 46
< Content-Type: application/json
```

while the body of the HTTP response will be

```
{  
  "status": 401,  
  "message": "Authenticate failed"  
}
```

## 1.7. Response formats

### 1.7.1. Introduction

The REST API can respond to the requests in different formats. By default, it returns a response in JSON format but it can also return a response in XML, CSV (for certain resources) and JS ([JSONP](#)) formats.

To change the format, .xml, .json, .csv or .js must be added to the end of the URI regardless of the HTTP verb (GET, POST, DELETE or PUT)

### 1.7.2. Examples

To retrieve the list of mailings in JSON format

```
GET /rest/resource.json HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QlLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded
```

To retrieve the list of mailings in XML format

```
GET /rest/resource.xml HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QlLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded
```

To retrieve the list of mailings in CSV format

```
GET /rest/resource.csv HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QlLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded
```

To retrieve the list of mailings in JSONP format

```
GET /rest/resource.js HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QlLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

callback=yourFunctionCallback
```

### 1.7.3. Cross-domain

Retrieving the response in javascript format allows you to overcome the difficulties linked to the cross-domain. Passing through a server in order to consult the API's directly is thus avoided. On the client side, it is recommended to use the jquery-jsonp plugin ([jQuery-jsonp on GitHub](#)) for error management (not initially available in JQuery).

For example, reading the resource of which the id is 2 via an ajax request returns

```
<script type="text/javascript" language="javascript" src="jquery.jsonp.js"></script>
<script>
$.jsonp({
  url: 'https://api.messengeo.net/rest/resource.js?callback=?',
  beforeSend: function (request) {
    request.setRequestHeader("Authorization", "Bearer " + ($("#accesstoken").val()));
  },
  data: {
    id: '2',
  }
}).done(function(data) {
  // data peut être un objectlist (size, total, list) ou une erreur (status, message)
  console.log(data);
}).fail(function(jqxhr, textStatus, errorThrown) {
  console.log('Errors occured');
});
</script>
```

## 1.8. Filters and passing multiple values

The read, update and delete actions use as input a filter which makes it possible to select only the resources to be read or to be affected. Most of these filters take several values.

There are two ways to pass these multiple values:

1. in the form of a character string with the values separated by commas;
2. in the form of a table.

For example, the following two requests allow you to retrieve the resources for which the is is equal either to 12, or equal to 13.

```
GET /rest/resource.json HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

id=12,13
```

```
GET /rest/resource.json HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: api.messengeo.net
Content-Type: application/x-www-form-urlencoded

id[0]=12&id[1]=13
```

## 1.9. Integrating our API as PHP

In order to simplify the integration of our REST APIs, we provide you with a library that facilitates the various calls from code written in php. This library is hosted on [GitHub](#).

## 2. API possibilities

### 2.1. Text message

Text messages are limited to 160 characters.

The following 10 characters are counted twice: ^\~[]{} , the euro symbol (€), and the Form Feed / FF character (0x0C). See appendix A for a list of characters supported in text mode.

### 2.2. Long text message

Can be used to send longer text messages, up to 1024 characters.

Long text messages are broken up into several short text messages, then reassembled on the user's mobile phone. Each text message contains 153 characters at most (the characters that are counted twice in short text messages are also counted twice here). If the mobile phone does not support long text messages, the message will appear as several messages.

The cost of a long text message is the sum of the cost of each short text message.

### 2.3. Binary text message

Binary mode lets you use any character in your text message. For example, you can use Chinese characters if you want to send a text message to China. A binary text message is composed of 70 characters.

### 2.4. International text messages

Can be used to send international text messages.

You must explicitly request that the "international" option be activated for your account. If you don't, text messages sent to a foreign mobile phone will not be delivered.

## 2.5. Mobile numbers

Mobile numbers must comply with the international format (e.g. +44 71234 56789)

The international country code for your country is defined when your Texteo account is created ("44" by default for the United Kingdom).

## 2.6. Scheduled text messages

You can schedule a text message to be sent at a later time. If no date is specified, the message will be sent immediately.

## 2.7. Volume

- **HTTPS API:** lets you send up to 200 text messages at once
- **SMTP, or FTP API:** lets you send up to 2,000 text messages at once (contact us if you need to send more than 2,000 messages at once)
- **Check most recent statuses:** up to 2,000

Note: text messages are sent at an average rate of 20 per second.



## 2.8. Text message statuses

The possible statuses of a text message are outlined in the table below:

Website indication	API code returned	Description
<b>Scheduled</b>	<b>wait</b>	The text message has been scheduled but not yet sent.
<b>In progress</b>	<b>on</b>	The text message has been sent but the recipient has not yet received it.
<b>Failed</b>	<b>ko</b>	The text message could not be sent. The reason for failure is given.
<b>Delivered</b>	<b>ok</b>	The text message has been delivered to its recipient.
<b>Answered</b>	<b>rep</b>	The text message has been delivered and the recipient has sent a reply.
<b>Unknown</b>	<b>no</b>	The final status of the text message is unknown.

In addition, a more specific technical code like "N5" or "N4:204" format is returned along with a more detailed description. These additional elements may change over time and are given for information purposes only.

## 2.9. Limitations

The status check frequency is limited. The limit is not currently categorically frozen but a moderate usage would be not to call more than 10 status requests per SMS sent.

So, it's also recommended:

- to group the status check into one request
- to use the status check of a mailing that returns the whole statuses of a mailing with only one request.
- to use the limit parameter
- not to check the status of an SMS that has already a final status (that cannot change)

If an abusive use is noticed, the status check could be temporarily disabled.

With one request, you can get up to 2,000 statuses.

## 2.10. Responses

As indicated above, replies to text messages are returned as a status.

In addition, responses can be sent to an e-mail address specific to your account. By default, the body of the e-mail message contains the text message that was sent along with its response, and the subject of the e-mail is "Response from 801 555 xxxx to SMS #xxx," where "801 555 xxxx" is the sender's mobile number and "#xxx" is the ID number of the corresponding text message. But you also have the possibility to change the subject and the sender of the email you will receive by filling the appropriate parameters (see parts 3.4).

Furthermore, you can change the format of the answer and choose between HTML and TEXT format, by contacting Digitaleo's support.

By default, your account is not configured to receive SMS responses. If you want to receive SMS responses, two possibilities has presented to you:

- You can contact Digitaleo's support to activate it for each SMS sent
- You can specify it when you create an SMS or a mailing with the **type** parameter. (type=2) - See [3.4. Sending a simple text message](#)

## 2.11. Ping

Instead of constantly requesting the status of a text message until it is received (polling), you can be alerted when status changes or when a reply is submitted to text messages you have sent.

For this, you provide an address on your server, and the Texteo platform will call this address each time there is an update, such as :

[http://www.serveur.com/ping.asp?sms\\_id=#sms\\_id#](http://www.serveur.com/ping.asp?sms_id=#sms_id#) or [http://www.serveur.com/#sms\\_id#](http://www.serveur.com/#sms_id#)  
#sms\_id# will be replaced by the text message ID.

Ping requests are done with HEAD http verb.

## 2.12. Optional parameters

Over the years, some optional parameters have been added to new API versions to clarify the use of the services.

Some optional parameters are not documented, they relate to special use cases of the API.

## 3. HTTPS API

### 3.1. HTTPS address

This is the internet link to the sms service

<https://sms.messenger.net/rest/>

### 3.2. Description of the sms resource

The sms resource allows you to send a text message to one contact only.

### 3.3. List of properties of the sms resource

Property	Description
id	Id of the message
status	Short status To have more informations on the status use the "status" resource.
text	Text of the message
date	Sending date
recipient	Recipient of the target
dialingCode	Dialing code of the SMS
email	Email address where the answers will be sent
emailSender	Sender of the response email
emailSubject	Subject of the response email

## 3.4. Sending a simple text message

Send a simple text message.

### 3.4.1. List of parameters to supply in order to send a simple text message

Parameters	Description
text	Text of the message to be sent
date	The date when the message as to be sent If there is no date or date in the past, the message will be sent immediately.
contacts	Table of contacts but with only one contact. A contact must have a field "recipient".
sender	This is the SMS header field that contains the message sender's number (also named TPOA). The TPOA field is limited by GSM standards to maximum 11 alphabet characters and digits. For legal reasons, Digitaleo has to approve each TPOA. So if you want to customize the sender id, please contact us.
email	Email address where the answers will be sent
emailSender	Sender of the response email
emailSubject	Subject of the response email
type	If you want to receive responses of this SMS, please specify this parameter with the value "2" (type=2) Careful! You can't ask to receive SMS responses and customize the sender of the SMS: theses options are incompatible. If they are both activated, the SMS won't be sent.

The parameters "text" and "contacts" are mandatory.

### 3.4.2. Example

```
POST /rest/sms HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbGciOiJSUz11NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
Content-Type: application/json

{
  "text": "Text message to be sent",
  "date": "2017-05-07 11:12:25",
```

```

"contacts":[
  {
    "civility":"M.",
    "firstName":"Jean",
    "lastName":"DUPONT",
    "recipient":"+33696300056"
  }
]
}

```

### 3.5. Sending a personalized text message

Send a personalized a text message.

#### 3.5.1. List of parameters to supply in order to send a personalized text message

Parameters	Description
text	Text of the message to be sent
date	The date when the message as to be sent If there is no date or date in the past, the message will be sent immediately.
contacts	Table of contacts but with only one contact. A contact must have a field "recipient".
email	Email address where the answers will be sent
emailSender	Sender of the response email
emailSubject	Subject of the response email
type	If you want to receive responses of this SMS, please specify this parameter with the value "2" (type=2)  Careful! You can't ask to receive SMS responses and customize the sender of the SMS: theses options are incompatible. If they are both activated, the SMS won't be sent.

The parameters "text" and "contacts" are mandatory.

#### 3.5.2. Example

```

POST /rest/sms HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
Content-Type: application/json

```

```

{
  "text": "Dear #civility# #firstName# #lastName#",
  "date": "2017-05-07 11:12:25",
  "contacts": [
    {
      "civility": "M.",
      "firstName": "Jean",
      "lastName": "DUPONT",
      "recipient": "+33696300056"
    }
  ]
}

```

## 3.6. Reading a text message

Read a text message.

### 3.6.1. List of available filters

Common filters

Filter	Description
id	Allows you to filter according to one or more sms ids
recipient	Allows you to filter according to one or more recipient
mailingName	Allows you to filter according to one or more name of mailing
mailingId	Allows you to filter according to one or more id of mailing
date	Allows you to filter according to one or more date
way	Allows you to filter according to you want to retrieve only SMS MT (Mobile transmitted : Digitaleo platform -> Mobile) or SMS MO (Mobile originated: Mobile -> Digitaleo platform ) Authorized values: <i>mt</i> or <i>mo</i>

### 3.6.2. Attributes for a text message returned by default

By default, only the id is returned. To access the other attributes, they must be specified in the properties parameter. Recall that the default attributes have an alias via the 'DEFAULT' keyword.

For example, to retrieve the default parameters and the recipient, the following must be specified

```

GET /rest/sms?properties=DEFAULT,recipient HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUz11NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net

```

### 3.6.3. Example

#### 3.6.3.1. Example 1: Retrieving all the message send to one phone number

```
GET /rest/sms?recipient=+33600000000 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

#### 3.6.3.2. Example 2: Retrieving the message sent to one phone number at a specified date

```
GET /rest/sms?recipient=+33600000001&date=2017-04-07 11:12:25 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

#### 3.6.3.3. Example 3: Retrieving all messages sent between 2 dates

To read all messages sent during a certain date range, you need to put it in an array. The format of the array is the following :

```
date[0][logical] = AND
date[1][operator] = >=
date[1][value] =2017-06-09
date[2][operator] = <
date[2][value] =2017-06-27
```

Which give us the following request:

```
GET /rest/sms? properties=DEFAULT & date[0][logical] = AND & date[1][operator] = >=&date[1][value] =
2017-06-09 & date[2][operator] = < &date[2][value] =2017-06-27 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

#### 3.6.3.4. Example 3: Retrieving all responses (SMS MO: from mobile to Digitaleo) of a mailing

```
GET /rest/sms?mailingId=123456&way=mo HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

### 3.7. Canceling a text message

Cancel a text message which has not been sent yet. Once a text message is canceled you can't read or access it anymore.

#### 3.7.1. List of available filters

Common filters

Filter	Description
id*	Allows you to delete according to one identifier

\*Mandatory

```
DELETE /rest/sms?id=1122334455 HTTP/1.1  
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...  
Host: sms.messengeo.net
```



## 3.8. Description of the mailing resource

A mailing allows you to send a template of text message to several contacts. You also have the possibility to personalize this message for each recipient.

### 3.8.1. List of properties of the mailing resource

Property	Description
id	Id of the mailing
name	Name of the mailing The name can be used to read the mailing or access its information later on.
text	Text of the message
date	Sending date
email	Email address where the answers will be sent
emailSender	Sender of the response email
emailSubject	Subject of the response email

### 3.8.2. List of parameters to supply in order to send a text message

Parameters	Description
text	Text of the message to be sent
name	Name of the mailing
date	The date when the message as to be sent If there is no date or date in the past, the mailing will be sent immediately.
contacts	Table of contacts. Each contact is in itself a table key/value. A contact must have a field "recipient".
sender	This is the SMS header field that contains the message sender's number (also named TPOA). The TPOA field is limited by GSM standards to maximum 11 alphabet characters and digits. For legal reasons, Digitaleo has to approve each TPOA. So if you want to customize the sender id, please contact us.
email	Email address where the answers will be sent

emailSender	Sender of the response email
emailSubject	Subject of the response email
type	If you want to receive responses of this SMS, please specify this parameter with the value "2" (type=2)  Careful! You can't ask to receive SMS responses and customize the sender of the SMS: these options are incompatible. If they are both activated, the SMS won't be sent.

The parameters "text" and "contacts" are mandatory.

### 3.8.3. Sending a simple text message mailing

```
POST /rest/mailing HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
Content-Type: application/json

{
  "text": "Text to be sent",
  "name": "Name of the mailing",
  "date": "2017-05-07 11:12:25",
  "contacts": [
    {
      "recipient": "+33696300056"
    },
    {
      "recipient": "+33696358756"
    }
  ]
}
```

### 3.8.4. Sending a personalized text message mailing

***This advanced mode can be used to send personalized text messages to several contacts.***

A contact is associated with a mobile number to which fields are attributed, such as "last name" and "first name." It is then possible to include these fields in the text message by enclosing them between pound signs (#).

Sample text message: "Dear #firstName# #lastName#, we would like to inform you..."

```
POST /rest/mailing HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
Content-Type: application/json

{
  "text": "Hello #firstName# #lastName#.",
  "name": "Name of the mailing",
  "date": "2017-05-07 11:12:25",
  "contacts": [
    {
      "firstName": "Jean",

```

```
"lastName":"DUPONT",
"recipient":"+33696300056"
},
{
"firstName":"Marc",
"lastName":"DUPOND",
"recipient":"+33696358756"
}
]
```

## 3.9. Reading a mailing

Read all the text message from a mailing.

### 3.9.1. List of available filters

Common filters

Filter	Description
id	Allows you to filter according to one or more mailing ids
name	Allows you to filter according to one or more names of mailing

### 3.9.2. Example

```
GET /rest/mailing?id=1234567890 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

## 3.10. Canceling a mailing

Cancel a mailing which has not been sent yet. Also works with a mailing in progress, it will cancel the messages not sent yet.

### 3.10.1. List of available filters

Common filters

Filter	Description
--------	-------------

id	Allows you to delete according to one identifier. This field is mandatory
----	--

```
DELETE /rest/mailing?id=198765432 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

### 3.11. Retrieving the current status of text message

The status resource allows you to check the state of a message, or several messages, telling you if the message is received, waiting, etc.

#### 3.11.1. List of properties of the status resource

Property	Description
smsId	Id of the SMS message
status	Short status To have more informations on the status use the "status" resource.
recipient	Recipient of the target
mailingId	Id of the mailing
date	Date of the status
description	Description of the status
response	If the status is "rep", this field contains the text of the response (text of the SMS MO : SMS sent by the mobile and received by the Digitaleo platform)

#### 3.11.2. Reading the status of one or several message

Read the status of one or more text messages with different filter.

##### 3.11.2.1. List of available filters

Common filters

Filter	Description
smsId	Allows you to filter according to one or more sms ids

mailingId	Allows you to filter according to one or more mailing ids
mailingName	Allows you to filter according to one or more names of mailing

### 3.11.2.2. Example 1: Read the current status of all messages from a mailing with its id.

```
GET /rest/status?mailingId=10002398 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

### 3.11.2.3. Example 2: Read the current status of a message.

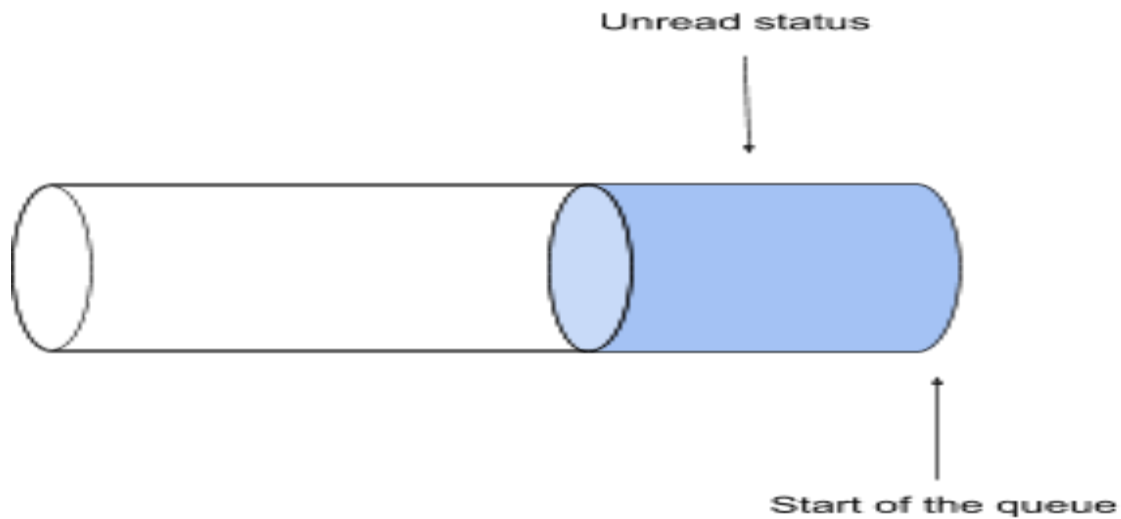
```
GET /rest/status?smsId=1234567890 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

### 3.11.3. Retrieving a queue of status

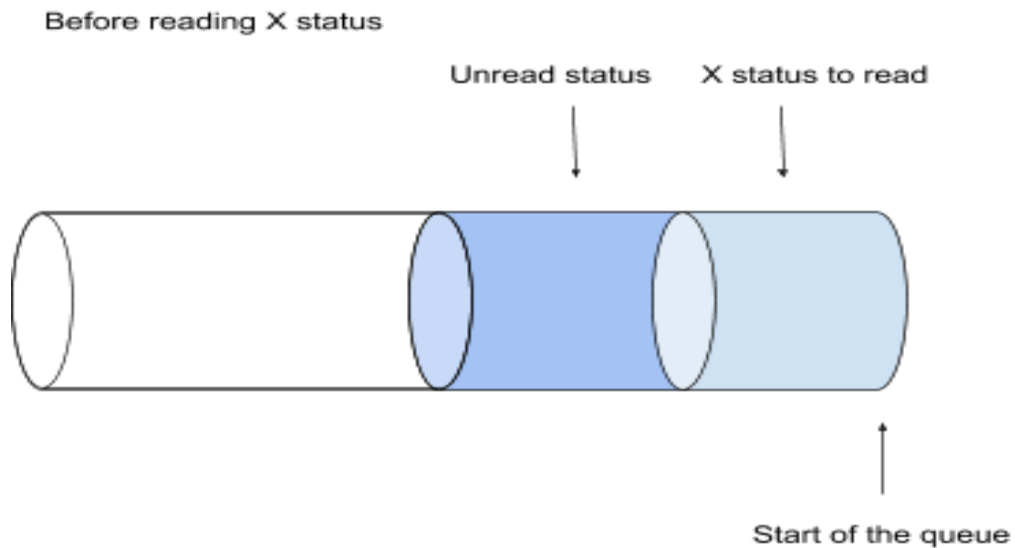
How the queue of status work

This queue works like a classical FIFO (First In First Out) queue.

New status, generated by any messages you sent, are added to the queue.

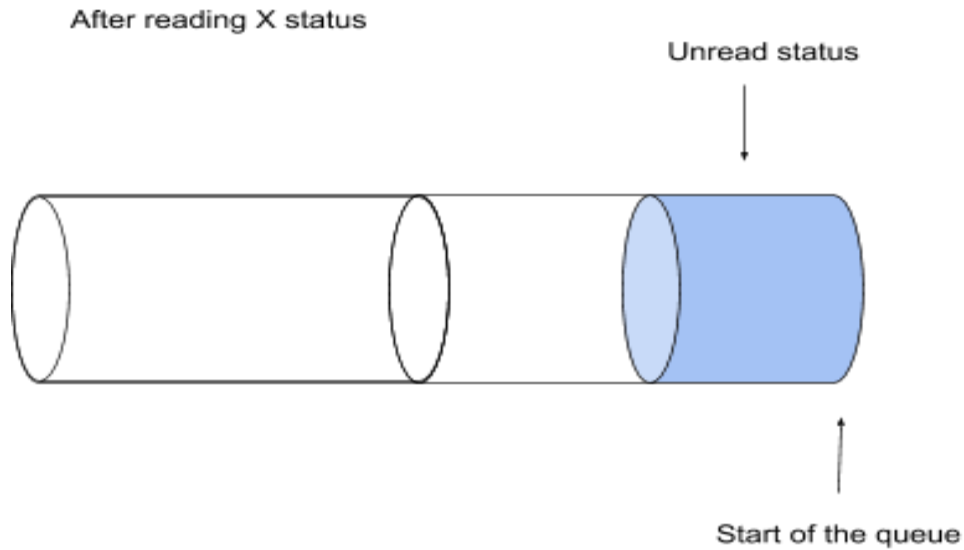


When you read the queue, you will read a list of status (you can choose the size of this list with the "limit" parameter or read all of them). This queue contains n status starting from the start of the queue, so the oldest status.

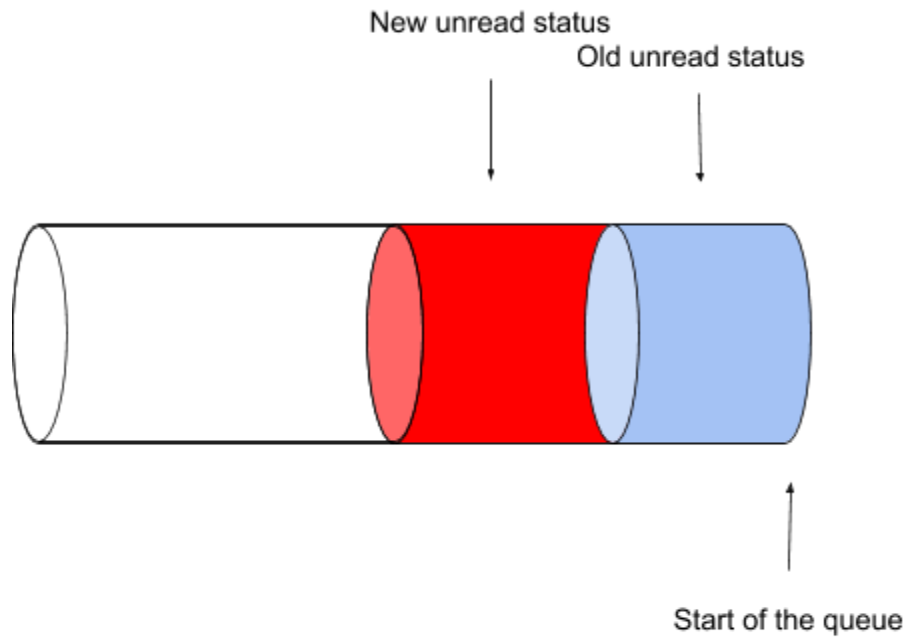


Every time you read a status, it will go out of the queue, and therefore will not be readable anymore by the queue

request (but you can still read it with the request describe in 3.15).



The queue will reduce like this until it's empty or until new status are added to the queue.



Every time you read the queue some (or all) status will be read and took out of the queue and every time a status is created it will be added to the queue.



This functionality can be useful if, for example, you want to follow the evolution of your messages. Instead of reading continuously the status of your messages you just have to check the queue to have all the evolution of your messages. Note that even if you read the queue of status you can still read the status of a message with the others requests. These other request are not impact by the usage of the “queue” request.

### 3.11.3.1. List of available filters

Filter	Description
limit	Number of status to retrieve

If no value is assigned to "limit" the function will retrieve all status from the pile.

### 3.11.3.2. Example 1: Retrieving all the status from the queue.

```
GET /rest/status?action=queue HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

### 3.11.3.3. Example 2: Retrieving the 4 last status from the queue.

```
GET /rest/status?action=queue&limit=4 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

## 3.12. Reading all the status of one text message

Read all the different status that one sms had.

### 3.12.1. List of properties of the smsstatus resource

Property	Description
smsId	Id of the SMS message
status	Short status To have more informations on the status use the "status" resource.
recipient	Recipient of the target
mailingId	Id of the mailing
date	Date of the status
description	Description of the status
response	If the status is "rep", this field contains the text of the response (text of the SMS MO : SMS sent by the mobile and received by the Digitaleo platform)

### 3.12.2. List of available filters

Filter	Description
smsId	Allows you to filter according to one or more sms ids

### 3.12.3. Example

GET request

```
GET /rest/smsstatus?smsId=1234567890&properties=smsId,status,description,date,response HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messenger.net
```

## Response

```
{
  "size":4,
  "total":null,
  "list":[
    {
      "smsId":"121905361",
      "status":"wait",
      "description":"SMS programmé",
      "date":"2017-09-25 17:41:00",
      "response":null
    },
    {
      "smsId":"121905361",
      "status":"ok",
      "description":"SMS reçu",
      "date":"2017-09-25 17:41:00",
      "response":null
    },
    {
      "smsId":"121905361",
      "status":"on",
      "description":"En cours",
      "date":"2017-09-25 17:41:01",
      "response":null
    },
    {
      "smsId":"121905361",
      "status":"rep",
      "description":"SMS répondu",
      "date":"2017-09-25 17:41:26",
      "response":"Yes, this open day can interest me"
    }
  ],
  "httpStatusCode":200
}
```

Here we can see all the different status a message had with the respective information (date, description, etc...) of each status.

### 3.13. Description of the stats resource

The statistic resource allows you to have all the statistics associated to the messages from one or several mailings.

#### 3.13.1. List of properties of the statistic resource

Property	Description
wait	Number of messages waiting to be send
ok	Number of messages received
on	Number of messages sent
no	Number of messages whose status is unknown
ko	Number of messages failed
rep	Number of messages with response
optout	Number of optout
total	Total number of messages

#### 3.13.2. Reading statistics

Read the statistics of a mailing

##### 3.13.2.1. List of available filters

Filter	Description
mailingId	Allows you to filter according to one or more mailing ids
mailingName	Allows you to filter according to one or more names of mailing
month	Allows you to filter according to a specific month

If no filter is given, this will return the statistics on all messages of the account.

### 3.13.2.2. Example 1: Retrieving the statistics using a mailing name

```
GET /rest/stats?mailingName=Mailing HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QilCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

### 3.13.2.3. Example 2: Retrieving the statistics from a month

To read all statistics from a specific month, you just need to specify the month. Like the following request :

```
GET /rest/stats?properties=DEFAULT&month=2017-06 HTTP/1.1
Authorization: Bearer eyJ0eXAIoiJKV1QilCJhbGciOiJSUzI1NiJ9.eyJpZCI6ImYyMzE2...
Host: sms.messengeo.net
```

Note: You can't use the mailingId/mailingName parameters with the month parameter.

## 4. SMTP API

### 4.1. E-mail address

To send your text message mailings, use the following SMTP API e-mail address:

```
sms@ecosms.fr
```

### 4.2. Sending a text message mailing

To send a text message mailing, just send an e-mail to sms@ecosms.fr with the following text (either in the message body or in an attachment):

```
U-code
D-2017-08-04T12:30:00
T-first text message for 3 recipients
#-447123456789
#-441234567891
#-441234567892
T-second text message
T-for 2 recipients
#-441234567893
#-441234567894
T-last text message for 3
#-441234567895
#-441234567896
#-441234567897
```

Where:

- A line beginning with U- containing the "code" of your texting account
- A line beginning with D- containing the sending date of the SMS.
  - You can set only one date for all SMS in the email.
  - This field is optional.
  - If not there, SMS will be sent immediately
  - If invalid, the email will not be taken into account.
  - The send date must respect the ISO-8601 format:
    - D-2017-08-04T15:19:21+02:00
    - D-2017-08-04T15:19:21
    - D-2017-08-04T15:19
    - D-2017-08-04
- One or more lines beginning with T- containing the text message (160 characters at most) to be sent to the recipients listed underneath
- One recipient per line beginning with #- (unlimited recipients)

- To send different text messages in a single e-mail, just alternate the messages in the text message fields (those that begin with "T-") and the recipients' mobile numbers (fields beginning with "#-"). Please note that the message is sent to the mobile numbers listed AFTER the message text.
- The subject of the e-mail and the name of the attachment do not matter.

### 4.3. Sending a customized text message mailing (advanced mode)

With this mode, you can send customized SMS towards several mobile recipients. Mobile numbers are completed with custom fields that can be included in the message.

For example:

```
#-447123456789;M;SMITH;John
#-441234567891;Mrs;JONES;Robert
```

Then, it's possible to include these fields in the SMS by respecting the following syntax: **#n#** where **n** is the position of the field after the mobile number.

For example

```
U-DEMO
T>Hello #1# #2#,
T-You car is ready.
T-Best regards
#-447123456789;M;SMITH
#-441234567891;Mrs;JONES
```

Will send the 2 following SMS

Hello M SMITH, You car is ready. Best regards	Hello Mrs JONES, You car is ready. Best regards
---	---

The following format is also authorized:

```
U-DEMO
T>Hello (%1) (%2),
T-You car is ready.
T-Best regards
#-447123456789;M;SMITH
#-441234567891;Mrs;JONES
```

## 4.4. PHP Example

This is the PHP code to send a text message mailing through the SMTP API:

```
<?php
$body = 'U-DEMO
T-first text message for 3 recipients
#-447123456789
#-441234567891
#-441234567892
T-second text message
T-for 2 recipients
#-441234567893
#-441234567894
T-last text message for 4
#-441234567895
#-441234567896
#-441234567897
#-441234567898;

mail( 'sms@ecosms.fr', "Send text message", $body );
```



## 5. FTP API

### 5.1. FTP address

This is the address of the FTP API server:

```
ftp://ftp.ecosms.fr
```

### 5.2. Sending a text message mailing

Contact us to obtain a username and password.

The format for files to be sent to the server is the same as for the SMTP API. See the previous section for details.

## 6. Texteo monitor website

### 6.1. Web address

This is the address of the Texteo Monitor website:

<https://monitor.texteo.net/>

### 6.2. Format

The Texteo Monitor website lets you monitor text messages sent by the following ecoSMS APIs: WSDL, HTTPS, SMTP, and FTP.

*Contact us to set up an account.*

# A. APPENDIX A - TEXT MESSAGE CHARACTERS

## List of supported characters for text messages in text mode

### 1. Single characters

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

## 2. Double characters

€ <FF> [ \ ]

^ { | } ~

Reference for the GSM 03.38 character set:

<http://www.unicode.org/Public/MAPPINGS/ETSI/GSM0338.TXT>

The following characters are changed when the message is sent:

- a with circumflex 'Â' => 'A', 'â' => 'a'
- e with circumflex 'Ê' => 'E', 'ê' => 'e'
- i with circumflex 'Î' => 'I', 'î' => 'i'
- o with circumflex 'Ô' => 'O', 'ô' => 'o'
- u with circumflex 'Û' => 'U', 'û' => 'u'
- e trema 'Ë' => 'E', 'ë' => 'e'
- i trema 'Ï' => 'I', 'ï' => 'i'
- Degree symbol '°' => 'o'
- Ligatured Œ 'œ' => 'oe'
- grave accent ' ` ' => ''
- smart apostrophe ' ’ ' => ''
- ellipsis points ' … ' => '...'

Characters not listed here are deleted.

## Copyright

All of this code is governed by French and international legislation on copyright and intellectual property. All reproduction rights reserved, including for documents that can be downloaded and iconographic and photographic representations. Reproducing all or a portion of this code on any support whatsoever is strictly forbidden unless authorization is obtained in writing from Digitaleo.