

MESSAGEFACTORY

SMS-MESSAGING API

CONTENTS

[API DOCUMENTATION \(SENDING SMS MESSAGES\).....6](#)

[1.MT SMS API.....6](#)

[Examples.....9](#)

API DOCUMENTATION (SENDING SMS MESSAGES)

1. MT SMS API

The MT SMS API provides functionality for sending SMS messages to one or more recipients. The call is triggered by either an HTTP GET or POST request to the following addresses:

<https://XXXXX.XX> (GET)

<https://XXXXX.XX> (POST)

In both the POST and GET variants, the following parameters can, and in some cases must be provided as parameters to the HTTP request (i.e. in the query string part of the request URL). These parameters must be latin-1 (ISO-8859-1) encoded. The parameters are:

Name	Alias	Notes	Description
l	login	Mandatory	Username (provided during account setup)
p	password	Mandatory	Password (provided during account setup)
msisdn	to	Mandatory	Message recipient. Multiple recipients can be specified by including multiple msisdn parameters in the request. Maximum amount of recipients per request is 500.
from	sender	Mandatory	Message originator. Can consist of max. 25 numbers or 11 alphanumeric characters (the originator field will be cut if the limit is exceeded)
dlr	-	Optional	True/false depending on whether delivery notifications are requested.

When triggered using HTTP GET, the request should also include the message content in the URL as follows:

Name	Alias	Notes	Description
msg	-	Mandatory when using the HTTP GET method.	The message (latin-1 encoded) and properly URL escaped.

The HTTP POST method reads the content from the request body, according to the content-type specified on the request. This makes it possible to submit characters that are not part of the latin-1 character set (e.g. the Euro sign). It is highly recommended that the HTTP POST variant is used (with UTF-8 encoding). All other parameters should be passed as request parameters (as described above), except the actual message payload. A Content-Type HTTP header should be set based on the type of content that is provided, and the actual content should be placed directly in the request body. The supported content-types are:

Content Type	Description
text/plain;charset="xxx"	Plain text encoded with the (optionally) provided character set. If the character set is omitted, ISO-8859-1 is assumed.

NOTE: The message content should be placed as such, encoded as indicated in the content-type, in the request body. Note that this is not how browsers generally submit POST data **special encoding** is used for the parameters, all of which are encoded in the request body).

The message payload can both in the POST and GET cases exceed the limits of one SMS message (e.g. 160 characters or 140 bytes). The gateway will automatically concatenate the messages, and report in its response how many parts were sent. All textual messages are reduced to match the GSM 03.38 alphabet if needed.

The SMS gateway will reply with an XML document that describes the result of the submission. In case of missing parameters, a reply with HTTP status code of 400 is returned (the cause can be found in the status text). The returned XML varies depending on whether delivery notifications are requested or not. It has the following DTD:

```
<!\ELEMENT delivery-report (accepted, failed?)>
<!\ELEMENT accepted (recipients, messages)>
<!\ELEMENT recipients (recipient?, #PCDATA)>
<!\ELEMENT recipient (#PCDATA)>
<!\ELEMENT messages (#PCDATA)>
<!\ELEMENT failed (msisdn+)>
<!\ELEMENT msisdn (#PCDATA)>
```

In the basic scenario (no delivery notifications), the number of recipients and messages and a list of possibly failed recipients is returned. For delivery notifications, each recipient and an associated message id is returned. This message id is passed back to the service provider when the delivery notification is returned (see below). Please see the examples for examples of how the gateway replies. In case of an internal error, a HTTP status code of 500 is returned. In this case the service provider should re-submit the message after a short while.

In case delivery notifications are enabled these are provided to the service provider through HTTP GET requests. During account creation, the service provider specifies an URL where delivery notifications should be provided. The HTTP GET request contains information about the status of the message (i.e. whether or not it was delivered). The request parameters that are added to the URL are:

Name	Description
id	message id, as provided when the message was submitted
msisdn	msisdn of the recipient
result	Either "delivered" when successful, or "failed" when failed. Also possible "accepted" when message is accepted for delivery and "deleted" when message is deleted

Examples

1) The message was delivered:

```
status = 200
<delivery-report>
  <accepted>
    <recipients>
      <recipient id="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx">[msisdn ]</recipient>
    </recipients>
    <messages>[length of all messages]</messages>
  </accepted>
</delivery-report>
```

2) The form of message is incorrect:

```
status = 400
<delivery-report>
  <failed>encode error </failed>
</delivery-report>
```

3) The form of parameter is incorrect:

```
status = 400
<delivery-report>
  <failed>parameters error ([parameter name] wrong format)</failed>
</delivery-report>
```

4) Message being sent is incorrect:

```
status = 400
<delivery-report>
  <failed>unknown error</failed>
</delivery-report>
```

5) Incorrect username or password:

```
status = 400
<delivery-report>
  <failed>login failed</failed>
</delivery-report>
```

Features:

1) It is possible to use same command ([MSISDN] POST or GET) to send message to several receivers.

```
<delivery-report>
  <accepted>
    <recipients>
      <recipient id="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx">[ 1 msisdn]</recipient>
      <recipient id="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx">[ 2 msisdn]</recipient>
      ....
    </recipients>
    <messages>[length of all messages]</messages>
  </accepted>
</delivery-report>
```