SMS OUT

a Geodrop guide

geodrop

SMS Out® 2.0

Geodrop API for SMS sending and controlling

Quick integration guide

RESTful API User Guide v1.3-EN, 21 September 2012

Contents

Changelog

Version	When	What	Who
1.0	Aug, 2nd 2012	First official version	Giuseppe Costa
1.1	Aug 17th 2012	Changed document name. Updated SMS delivery status codes table. Updated error codes table for SendSMS method in Appendix A.	Giuseppe Costa
1.2	Aug 27th 2012	Minor changes in text	Stefano di Sandro
1.3	Sept 21th 2012	Minor changes in text	Giuseppe Costa

Abstract

This guide provides useful information for a quick set up of a system enable to receive SMS messages via "SMS Out" API module.

The description of the data sent and received via the API is provided using *XML Schema* (http://www.w3.org/XML/Schema), a W3C standard formalism for the schematic definition of XML documents.

To build the examples we used the cURL utility (http://curl.haxx.se/).

System overview

This chapter is intended to provide an overview of the various components of the module for sending SMS messages, without going into details, in order to enable the reader to understand roughly how it works.

Sending messages

Geodrop can send SMS messages to mobile phones through a set of dedicated connections with the mobile operators.

To be able to send SMS messages through Geodrop you need only to have a positive balance in your account.

You can send SMS messages directly or schedule to send on a specific date and time.

For the default cost of SMS messages, please refer to the “Pricing” section of the Geodrop website. If you need to agree on custom rates, please contact the sales support (sales@geodrop.com).

Scheduled delivery

If you choose to use scheduled delivery, Geodrop will pass the message to the operator only on the date and time you specify, so it will take care of the actual delivery to recipients.

Once Geodrop has accepted your request, it stores internally all the information needed to send the message and waits the date and time requested to pass the message to the operator.

We call **job** the set of information stored by Geodrop, necessary for the subsequent sending of the message.

When Geodrop starts passing the message to the operator, we say that it is **executing** the job. A job waiting to be executed is called **scheduled**.

A job is **completed** when all the informations has been passed to the operator

You can modify a job up to 5 minutes before the scheduled time, changing every aspect or completely deleting it.

Immediate delivery

Using immediate delivery Geodrop will immediately create the corresponding job and execute it. Once Geodrop has accepted your request, you cannot anymore stop or change the dispatch of messages.

Delivery report

As messages are processed by operators, they send notifications back to Geodrop for each recipient in which they communicate the successful delivery or failure. These notification are called **delivery reports** (shortened: **DLR**). DLR are asynchronous and may be returned to Geodrop up to 5 days after

sending the messages.

To each recipient of a message, Geodrop associates a **state of delivery.** See *Table 1* on page 13 for a full list of the states of delivery.

The overall **status** of a job is given by the set of states associated to individual recipients of the corresponding message. As the DLR arrives, the job's overall status changes.

Message archive

A job (and the corresponding messages) is **archived** if one of the following conditions is true:

- Geodrop has received a DLR for each one of the message recipients;
- Geodrop has not received all DLR, but 5 days have elapsed since the message was passed to the operator.

An archived job will never change its overall status.

How to send SMS messages

To send SMS messages through the API of the SMS Out module you only need to have a positive balance in your account and to have an active application linked to your geodrop profile.

To activate an application, you must select one from the Geodrop Market or create your own one becoming a developer.

To join the Geodrop developers program, please follow the “Developers” link in the website footer, under the “Community” link group.

Checking your balance

To check your current balance you must call the **profile** method in http POST as shown in the next example:

```
curl -X POST -H "Authorization: Bearer <access-token>" -H "Content-type: application/x-www-form-urlencoded" https://api.geodrop.net/out/1/users/1/profile
```

The “<access-token>” is a 32 hexadecimal digit string used in the Oauth2 authentication and authorization process. To know how to get the access token we remand to the Oauth2 documentation on Geodrop website.

In response to the https request shown above, you will receive an XML document like the following:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEORESPONSE><USER userid="<ID>" sent="0" credit="<Balance>" account="Prepaid" expire="2025-09-30"/></GEORESPONSE>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEORESPONSE>
 <USER userid="<ID>" sent="0" credit="<Balance>" account="Prepaid" expire="2025-09-30"/>
```

```
</GEORESPONSE>
```

See “Get user profile, Response” in Appendix B for a complete XML Schema Definition (from now *XSD*) of the response.

The response contains only the tag **USER** with the following attributes:

- **userid**: your own Geodrop user identifier;
- **credit**: your account balance in millionth of euro;
- **account**: account type. Possible value are “Postpaid” and “Prepaid”. “Prepaid” accounts are freely activable from Geodrop website and require that the credit is recharged by the user.To activate “Postpaid” accounts please contact the sales support (sales@geodrop.com);
- **expire**: account expire date.

In case of error, the **USER** tag is replaced by the **BAD** tag, containing the error code and a human readable description of the problem. Here is an example:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEORESPONSE><BAD code="100"
description="METHOD NOT ALLOWED"/></GEORESPONSE>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEORESPONSE>
 <BAD code="100" description="METHOD NOT ALLOWED"/>
</GEORESPONSE>
```

Estimating message cost

Before actually sending a message, you can ask for a “quote”. According to the mobile phone numbers of the recipients and the message length, Geodrop is able to determine a higher estimate of the final cost of the message.

To get the quote you must call the **estimatecost** method in http POST, specifying a text for the message and a list of recipients. Here is a request schema for this method:

```
curl -X POST -H "Authorization: Bearer <access-token>" --data '<request-body>'
https://api.geodrop.net/out/1/sms/1/estimatecost
```

The`<request-body>` must conform to the XSD specified in Appendix B, “Estimate message cost, Request body”.

A sample `<request-body>` to estimate the cost of sending a message to two recipients is shown below:

Raw

```
<?xml version="1.0" encoding="UTF-8"?><GEOSMS><MESSAGE content="text"><TEXT><!
[CDATA[This SMS is powered by Geodrop]]></TEXT></MESSAGE><LIST><DEST
msisdn="+393471234567" /><DEST msisdn="+393351234568" /></LIST></GEOSMS>
```

Indented

```
<?xml version="1.0" encoding="UTF-8"?>
<GEOSMS>
 <MESSAGE content="text">
  <TEXT><![CDATA[This SMS is powered by Geodrop]]></TEXT>
 </MESSAGE>
 <LIST>
  <DEST msisdn="+393471234567" />
  <DEST msisdn="+393351234568" />
 </LIST>
</GEOSMS>
```

The message text is specified as the content of the **TEXT** tag. It is strongly recommended to enclose the text in a **CDATA** section (like in the sample above) to avoid problems with special characters.

Each tag **DEST** contains a single recipient as the value of its **msdidn** attribute.

The Geodrop response is a XML document complying with the XSD specified in Appendix B, “Estimate message cost, Response”.

The estimate cost for the above request is shown in the following example:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?
><GEORESPONSE><ESTIMATECOST>160000</ESTIMATECOST><USER userid="<ID>" sent="0"
credit="<balance>" account="Prepaid"/></GEORESPONSE>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEORESPONSE>
 <ESTIMATECOST>160000</ESTIMATECOST>
 <USER userid="<ID>" sent="0" credit="<balance>" account="Prepaid"/>
</GEORESPONSE>
```

The estimated final cost is enclosed by the **ESTIMATECOST** tag.

The **USER** tag is similar to that returned by the **profile** method. Compared with the response of this method, the attribute **expire** is missing.

Sending messages

The simplest way to send SMS messages is to call the **send** method in http POST, specifying a text for the message, a list of recipients and a personalized sender.

```
curl -X POST -H "Authorization: Bearer <access-token>" --data '<request-body>'
https://api.geodrop.net/out/1/sms/1/send
```

The`<request-body>` must conform to the XSD specified in Appendix B, “Send SMS, Request”.

Immediate dispatch

The XML document to immediately send a message to three recipients is a similar to that used for the estimation of the cost. A sample is shown below:

Raw

```
<?xml version="1.0" encoding="UTF-8"?><GEOSMS><MESSAGE content="text"><TEXT><!
[CDATA[This SMS is powered by Geodrop]]></TEXT></MESSAGE><LIST><DEST
msisdn="+393471234567" /><DEST msisdn="+393351234568" /><DEST msisdn="aaaa"
/></LIST><TPOA>GEODROP</TPOA></GEOSMS>
```

Indented

```
<?xml version="1.0" standalone="yes"?>
<GEOSMS>
 <MESSAGE content="text">
  <TEXT><![CDATA[This SMS is powered by Geodrop]]></TEXT>
 </MESSAGE>
 <LIST>
  <DEST msisdn="+393471234567" />
  <DEST msisdn="+393351234568" />
  <DEST msisdn="aaaa" />
 </LIST>
 <TPOA>GEODROP</TPOA>
</GEOSMS>
```

The request body has the same pattern as that of the method **estimatecost**, except for the **TPOA** tag, used to specify the personalized sender.

Geodrop will send you back a response containing a first level report providing the following informations:

- the total of the requested recipients and of the effectively sent SMS messages. Those values are reported respectively in the **requested** and **posted** attributes of the **REPORT** tag;
- for each recipient's mobile phone number:
  - if it has been accepted, it is reported as the **msisdn** attribute of an **OK** tag;
  - if it has been rejected, it is reported as the **msisdn** attribute of an **ERROR** tag.

The response to the above request is shown in the following example:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEORESPONSE><REPORT
requested="3" posted="2"><OK orderid="8D675B2D-8BF4-43BB-86DD-A6315AB4A721"
msisdn="+393341117125" /><OK orderid="8D675B2D-8BF4-43BB-86DD-A6315AB4A721"
msisdn="+391111111111" /><ERROR reason="1" msisdn="aaaa" /></REPORT><USER userid="<ID>"
sent="0" credit="<balance>" account="Prepaid" /></GEORESPONSE>
```

Indented

```
<?xml version="1.0" standalone="yes"?>
<GEORESPONSE>
 <REPORT requested="3" posted="2">
  <OK orderid="8D675B2D-8BF4-43BB-86DD-A6315AB4A721" msisdn="+393341117125" />
  <OK orderid="8D675B2D-8BF4-43BB-86DD-A6315AB4A721" msisdn="+391111111111" />
  <ERROR reason="1" msisdn="aaaa" />
 </REPORT>
```

```
 <USER userid="<ID>" sent="0" credit="<balance>" account="Prepaid" />
</GEORESPONSE>
```

The **USER** tag is the same as the **estimatecost** method.

Delayed dispatch

To schedule the dispatch of a message to a certain date, we simply need to add the **DEFERRED** tag to the XML used to send the message immediately. Within the tag are given the date and time for sending the message.

Geodrop will process your request, sending back the cost of the message and the first level report containing the list of accepted and rejected recipients, but will store your message waiting the requested date and time to actually send it.

To send the message of the previous example on *December, 12 2012 at 12:00*, we need to modify the request body, adding the appropriate **DEFERRED** tag as follow:

Raw

```
<?xml version="1.0" encoding="UTF-8"?><GEOSMS><MESSAGE content="text"><TEXT><!
[CDATA[This SMS is powered by Geodrop]]></TEXT></MESSAGE><LIST><DEST
msisdn="+393471234567" /><DEST msisdn="+393351234568" /><DEST msisdn="aaaa"
/></LIST><TPOA>GEODROP</TPOA><DEFERRED>2012-12-12 12:00</DEFERRED></GEOSMS>
```

Indented

```
<?xml version="1.0" encoding="UTF-8"?>
<GEOSMS>
 <MESSAGE content="text">
  <TEXT><![CDATA[This SMS is powered by Geodrop]]></TEXT>
 </MESSAGE>
 <LIST>
  <DEST msisdn="+393471234567" />
  <DEST msisdn="+393351234568" />
  <DEST msisdn="aaaa" />
 </LIST>
 <TPOA>GEODROP</TPOA>
 <DEFERRED>2012-12-12 12:00</DEFERRED>
</GEOSMS>
```

The response will be identical to the case of immediate sending.

Managing sent and scheduled messages

Once Geodrop has accepted your requests to send SMS (immediately or deferred), you can get informations on the corresponding jobs through a set of dedicated methods.

For **completed** and **archived** jobs you can get the overall status and details of which recipients received the message and which did not.

For **scheduled** jobs you have more available actions. You can:

- get the job status, i.e. its description (text and recipient's list) and the date and time planned for sending;
- modify the text, the recipient's list and the planned date and time;
- delete the job (the message will not be sent anymore).

Retrieving completed and archived job status

To retrieve the overall status of a completed or archived job you must call the **status** method in http PUT. The request schema is shown below:

```
curl -X PUT -H "Authorization: Bearer <access-token>" -H "Content-type: application/xml"
--data '<request-body>' https://api.geodrop.net/out/1/sms/1/status
```

The`<request-body>` must conform to the XSD specified in Appendix B, “Retrieve SMS status, Request body”.

A sample `<request-body>` to get the status of a specified job is shown below:

Raw

```
<?xml version="1.0" standalone="yes"?
><GEOSMSSTATUS><ID_CLIENT>xxx</ID_CLIENT><REQUEST_TYPE>job</REQUEST_TYPE><JOB
orderid="<job-id>" limit="0,100"/></GEOSMSSTATUS>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEOSMSSTATUS>
 <ID_CLIENT>xxx</ID_CLIENT>
 <REQUEST_TYPE>job</REQUEST_TYPE>
 <JOB orderid="<job-id>" limit="0,100" />
</GEOSMSSTATUS>
```

The value given in the optional **ID_CLIENT** tag is an alphanumeric string of any kind and will be sent back from Geodrop in the response, enclosed in the same tag. It is designed for development needs.

The value “job” passed in the **REQUEST_TYPE** tag tells Geodrop that you are requesting informations for a whole job. The status method can be used to get informations about the status of any set of SMS messages: from a single SMS to all those available in archive. Accepted values for this tag are:

- **job**: used to retrieve the overall status of a job. This request type requires the **JOB** tag (with the *Global Unique IDentifier* (GUID) of the job) to be specified within the request body;
- **range**: used to retrieve the status of all SMS messages sent in a given time interval. The date range must be specified as the attribute of the **RANGE** tag;
- **adhoc**: used to retrieve the status of any set of SMS messages. The set of messages is specified in the tag **LIST** through a series of tags **DEST**, each of which contains the pair *<orderid, MSISDN>* that uniquely identifies a single SMS message in the Geodrop archive.

For the complete syntax of each request type, please refere to the XSD in Appendix B, “Retrieve SMS status, Request body”.

The **JOB** and the **RANGE** tag have both a **limit** attribute, used to paginate the results. The limit value consists of two integers separated by a comma. The first one indicates the position of the first required result and the second the total number of results to return.

You can get at most 1000 results at a time, so to retrieve the overall status of a job with more than 1000 recipients you must do the following steps:

- do a first request with limit 0,1000
- do a second request with limit 1001, 1000
- go on like this, until you retrieve all the informations you need.

The response for any one of the request types listed above has always the same format, described by the XSD specified in Appendix B, “Retrieve SMS status, Response”.

The response corresponding to the request of the previous example is reported below:

Raw

```
<?xml version="1.0" standalone="yes"?> <GEOSMSSTATUSRESPONSE> <ID_CLIENT>xxx</ID_CLIENT>
<REQUEST_TYPE>job</REQUEST_TYPE> <REPORT requested="2" posted="2"><OK state="DELIVERED"
msisdn="+393341117125" orderid="<job-id>"/><ERROR state="NOT DELIVERED"
msisdn="+3911111" orderid="<job-id>"/><OK state="ACCEPTED" msisdn="+391111111111"
orderid="<job-id>"/></REPORT></GEOSMSSTATUSRESPONSE>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEOSMSSTATUSRESPONSE>
 <ID_CLIENT>xxx</ID_CLIENT>
 <REQUEST_TYPE>job</REQUEST_TYPE>
 <REPORT requested="3" posted="3">
  <OK state="DELIVERED" msisdn="+393341117125" orderid="<job-id>" />
  <ERROR state="NOT DELIVERED" msisdn="+3911111" orderid="<job-id>" />
  <OK state="ACCEPTED" msisdn="+391111111111" orderid="<job-id>" />
 </REPORT>
</GEOSMSSTATUSRESPONSE>
```

The **REPORT** tag has two attributes:

- **requested**: the total number of SMS messages requested;
- **posted**: the real number of SMS messages returned.

For each recipient of the requested job, you will find a dedicated tag within the report. In detail:

- **ERROR** tag for recipients who have certainly not received the message;
- **OK** tag for all other cases.

Both tags have the following attributes:

- **msisdn**: the recipient's mobile phone number;

- **orderid**: the job GUID;
- **state**: indicates if the recipient has received the SMS message for the job corresponding to the **orderid**. Possible values are shown in Table 1.

Status	Description
UNKNOWN_ERROR	Unknown error. Please contact the technical support
ERROR_INTERNAL	Message not sent due to an internal error
NOT_ENOUGH_BALANCE	Message not sent to the recipient because the user has not enough balance
ACCEPTED	The operator has not returned any DLR for the SMS message sent to the recipient, so Geodrop cannot state whether it has been delivered or not
DELIVERED	The recipient has received the SMS message (a confirmation DLR has been received from the operator)
NOT_DELIVERED	The recipient has not received the SMS message (a failure DLR has been received from the operator)
NOT_FOUND	No status informations available for the recipient
UNAUTHORIZED	User is not authorized to receive this information

Table 1: possibile values for the state attribute

Managing scheduled jobs

To manage scheduled jobs you must call the **scheduled** method in http POST. The request schema is shown below:

```
curl -X POST -H "Authorization: Bearer <access-token>" -H "Content-type:
application/xml" --data '<request-body>'
https://api.geodrop.net/out/1/sms/1/jobs/scheduled
```

The`<request-body>` must conform to the XSD specified in Appendix B, “Manage scheduled jobs, Request body”.

This method performs different actions depending on the value passed in the **action** attribute of the **JOB** tag. Specific cases are detailed in the following paragraphs.

The response for any one of the possible actions has always the same format, described by the XSD specified in Appendix B, “Retrieve SMS status, Response)”. It contains only a **JOB** tag with the following attributes:

- **jobid**: the GUID of the job involved in the transaction;
- **transaction**: “OK” or “KO” values, indicating respectively the global success or failure of the transaction;
- **cause**: a short description of any problems;
- **errorcode**: the internal code of the error if any.

Get scheduled jobs status

To retrieve the overall status of a scheduled job the action parameter must contain the value “STATUS”, like in the following example:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEOSMSSCHEDULER><JOB
JOBID="<job-id>" ACTION="STATUS"></JOB></GEOSMSSCHEDULER>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEOSMSSCHEDULER>
 <JOB JOBID="<job-id>" ACTION="STATUS"></JOB>
</GEOSMSSCHEDULER>
```

The **<job-id>** is the GUID of the job for which we are requesting the global status.

The response will be like the following:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEOSMSSCHEDULER><JOB
JOBID="<job-id>" TRANSACTION="OK" CAUSE="DONE" CODE="2" /><DEFERREDTIME>2012-12-12
12:00:00</DEFERREDTIME><MESSAGETEXT><![CDATA[This SMS is powered by
Geodrop]]></MESSAGETEXT><TPOA><![CDATA[GEODROP]]></TPOA></GEOSMSSCHEDULER>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEOSMSSCHEDULER>
 <JOB JOBID="<job-id>" TRANSACTION="OK" CAUSE="DONE" CODE="2" />
 <DEFERREDTIME>2012-12-12 12:00:00</DEFERREDTIME>
 <MESSAGETEXT><![CDATA[This SMS is powered by Geodrop]]></MESSAGETEXT>
 <TPOA><![CDATA[GEODROP]]></TPOA>
</GEOSMSSCHEDULER>
```

Modify scheduled jobs

You can modify the following aspects of a scheduled job:

- the message text;
- the personalized sender;
- scheduled date and time;
- recipients list.

Assuming you want to update your message as follow:

- change the text: “**Geodrop brings your SMS to the highest level!**”;

- change the sender: “**GEODROPTEAM**”;
- set the scheduled time to **12:12:12**;
- remove the recipient having the mobile phone number “**+393471234567**”;
- add the mobile phone number “**+393333333333**” as a new recipient;

the body of your request will be the following:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEOSMSSCHEDULER><JOB
JOBID="<job-id>" ACTION="MODIFY"><MESSAGETEXT><![CDATA[Geodrop brings your SMS to the
highest level!]]></MESSAGETEXT><LISTMSISDN><DEL msisdn="+393471234567" /><ADD
msisdn="+393333333333" /></LISTMSISDN><TPOA>GEODROPTEAM</TPOA><DEFERREDTIME>2012-12-12
12:12:12</DEFERREDTIME></JOB></GEOSMSSCHEDULER>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEOSMSSCHEDULER>
 <JOB JOBID="<job-id>" ACTION="MODIFY">
  <MESSAGETEXT><![CDATA[Geodrop brings your SMS to the highest level!]]></MESSAGETEXT>
  <LISTMSISDN>
   <DEL msisdn="+393471234567" />
   <ADD msisdn="+393333333333" />
  </LISTMSISDN>
  <TPOA>GEODROPTEAM</TPOA>
  <DEFERREDTIME>2012-12-12 12:12:12</DEFERREDTIME>
 </JOB>
</GEOSMSSCHEDULER>
```

The response will report only the success or the failure of the requested updates, as in the following example (successful update).

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEOSMSSCHEDULER><JOB
JOBID="<job-id>" TRANSACTION="OK" CAUSE="DONE" ERRORCODE="2" /> </GEOSMSSCHEDULER>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEOSMSSCHEDULER>
 <JOB JOBID="<job-id>" TRANSACTION="OK" CAUSE="DONE" ERRORCODE="2" />
</GEOSMSSCHEDULER>
```

Delete scheduled jobs

To delete a scheduled job you need to specify the job GUID and set the action to “DELETE”. Here is a sample request body:

Raw

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><GEOSMSSCHEDULER><JOB
```

```
JOBID="<job-id>" ACTION="DELETE"></JOB></GEOSMSSCHEDULER>
```

Indented

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GEOSMSSCHEDULER>
 <JOB JOBID="<job-id>" ACTION="DELETE"></JOB>
</GEOSMSSCHEDULER>
```

The response will report only the success or the failure of the requested action, as shown for the modification of the job.

Appendix A: status codes and charset

Send and estimatecost methods

BAD tag status codes

Status	Code
DO_SEND_SMS_XML_MALFORMED	2
DO_SENDSMS_DEFERRED_TIME_DATE_OUT	603
DO_SENDSMS_DEFERRED_TIME_DATE_TOO_OLD	602
DO_SENDSMS_DEFERRED_TIME_BAD	601
DO_SENDSMS_MESSAGE_TYPE_UNKNOWN	502
DO_SENDSMS_MESSAGE_NO_TYPE	501
DO_SENDSMS_TEXT_NOT_VALID	403
DO_SENDSMS_TEXT_TOO_LONG	402
DO_SENDSMS_TEXT_EMPTY	401
DO_SENDSMS_MSISDN_LIST_EMPTY	301
DO_SENDSMS_MSISDN_LIST_TOO_LONG	302
DO_SENDSMS_DROP_UNAVAILABLE	201
DO_SENDSMS_MESSAGE_MALFORMED	104
DO_SENDSMS_INTERNAL_ERROR	101
DO_SENDSMS_INVALID_RUN_RANGE	105
DO_API_SENDSMS_INVALID_SENDER	107
DO_SENDSMS_INVALID_TEMPLATE	108
DO_SENDSMS_INVALID_NOTIFY_URL	109

Scheduler status code

Status	Code
DO_JOBSCHEDULED_XML_MALFORMED	0x01
DO_JOBSCHEDULED_XML_OK	0x02
DO_JOBSCHEDULED_DEFERRED_TIME_DATE_OUT	0x03
DO_JOBSCHEDULED_DEFERRED_TIME_DATE_TOO_OLD	0x04
DO_JOBSCHEDULED_DEFERRED_TIME_BAD	0x05
DO_JOBSCHEDULED_TEXT_NOT_VALID	0x06
DO_JOBSCHEDULED_TEXT_TOO_LONG	0x07
DO_JOBSCHEDULED_TEXT_EMPTY	0x08
DO_JOBSCHEDULED_MSISDN_LIST_EMPTY	0x09
DO_JOBSCHEDULED_MSISDN_LIST_TOO_LONG	0x0A
DO_JOBSCHEDULED_DROP_UNAVAILABLE	0x0B

Status	Code
DO_JOBSCHEDULED_MESSAGE_MALFORMED	0x0C
DO_JOBSCHEDULED_INVALID_ACTION	0x0D
DO_JOBSCHEDULED_INVALID_JOBID	0x0E
DO_JOBSCHEDULED_INVALID_TPOA	0x0F
DO_JOBSCHEDULED_INVALID_TID	0x10
DO_JOBSCHEDULED_INTERNAL_ERROR	0x11
DO_JOBSCHEDULED_JOBNOTFOUND	0x12
DO_JOBSCHEDULED_INVALID_RUN_RANGE	0x13
DO_JOBSCHEDULED_INVALID_NOTIFY_URL	0x14

GSM 03.38 charset

3GPP TS 23.038 / GSM 03.38																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	@	£	$	¥	è	é	ù	ì	ò	Ç	*LF*	Ø	ø	*CR*	Å	å
1x	Δ	_	Φ	Γ	Λ	Ω	Π	Ψ	Σ	Θ	Ξ	*ESC*	Æ	æ	ß	É
2x	*SP*	!	"	#	¤	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	¡	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ñ	Ü	§
6x	¿	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ñ	ü	à
1B 0x											*FF*					
1B 1x					^							*ESC2*				
1B 2x									{	}						\
1B 3x													[	~	]	
1B 4x	|															
1B 5x																
1B 6x						€										
1B 7x																

Appendix B: XML Schema Definitions

Get user profile

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define user_t type -->
 <xs:complexType name="user_t">
  <xs:attribute name="USERID" use="required" type="xs:string"/>
  <xs:attribute name="SENT" use="required" type="xs:nonNegativeInteger"/>
  <xs:attribute name="CREDIT" use="required" type="xs:decimal"/>
  <xs:attribute name="ACCOUNT" use="required" type="xs:string"/>
  <xs:attribute name="EXPIRE" use="required" type="xs:string"/>
 </xs:complexType>
 <!-- end of user_t type -->

 <!-- define userprofilebad_t type -->
 <xs:complexType name="userprofilebad_t">
  <xs:attribute name="CODE" use="required" type="xs:nonNegativeInteger"/>
  <xs:attribute name="DESCRIPTION" use="required" type="xs:string"/>
 </xs:complexType>
 <!-- end of userprofilebad_t type -->

 <!-- define georesponse_t type -->
 <xs:complexType name="georesponse_t">
  <xs:sequence>
   <xs:element name="USER" type="user_t" maxOccurs="1" minOccurs="0"/>
   <xs:element name="BAD" type="userprofilebad_t" maxOccurs="1" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
 <!-- end georesponse_t type -->

 <!-- Define completely the message from complex type -->
 <xs:element name="GERESPONSE" type="georesponse_t"/>

</xs:schema>
```

Estimate message cost

Request body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define MESSAGE type -->
 <xs:complexType name="message_t">
  <xs:sequence>
   <xs:element name="TEXT" type="xs:string" maxOccurs="1" />
  </xs:sequence>
  <xs:attribute name="content" use="required" type="xs:string" fixed="text" />
 </xs:complexType>
 <!-- end of MESSAGE type -->

 <!-- define DEST type -->
 <xs:complexType name="dest_t">
  <xs:attribute name="msisdn" use="optional" type="xs:string" />
  <xs:attribute name="contactid" use="optional" type="xs:nonNegativeInteger" />
  <xs:attribute name="groupid" use="optional" type="xs:nonNegativeInteger" />
  <xs:attribute name="bookname" use="optional" type="xs:string" />
 </xs:complexType>
 <!-- end DEST type -->

 <!-- define LIST type -->
 <xs:complexType name="list_t">
  <xs:sequence>
   <xs:element name="DEST" minOccurs="1" maxOccurs="unbounded" type="dest_t" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of LIST type -->

 <!-- define GEOSMS Type -->
 <xs:complexType name="geosms_t">
  <xs:sequence>
   <xs:element name="MESSAGE" type="message_t" maxOccurs="1" />
   <xs:element name="LIST" type="list_t" maxOccurs="1" />
   <xs:element name="TPOA" type="xs:string" maxOccurs="1" minOccurs="0" />
   <xs:element name="DEFERRED" type="xs:string" maxOccurs="1" minOccurs="0" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of GEOSMS Type -->

 <!-- Define completely the message from complex type -->
```

```
 <xs:element name="GEOSMS" type="geosms_t" />

</xs:schema>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define USER type -->
 <xs:complexType name="user_t">
  <xs:attribute name="userid" use="required" type="xs:string" />
  <xs:attribute name="sent" use="required" type="xs:nonNegativeInteger" />
  <xs:attribute name="credit" use="required" type="xs:decimal" />
  <xs:attribute name="account" use="required" type="xs:string" />
  <xs:attribute name="expired" use="required" type="xs:string" />
 </xs:complexType>
 <!-- end of USER type -->

 <!-- define GORESPONSE Type -->
 <xs:complexType name="georesponse_t">
  <xs:sequence>
   <xs:element name="ESTIMATECOST" type="nonNegativeInteger" maxOccurs="1" minOccurs="1"
/>
   <xs:element name="USER" type="user_t" maxOccurs="1" minOccurs="1" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of GORESPONSE Type -->

 <!-- Define completely the message from complex type -->
 <xs:element name="GORESPONSE" type="georesponse_t" />

</xs:schema>
```

Send SMS

Request body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define RUNRANGE type -->
 <xs:complexType name="runrange_t">
  <xs:sequence>
   <xs:element name="L_TIME" type="xs:string" maxOccurs="1" minOccurs="1" />
   <xs:element name="TIME_WINDOW" type="nonNegativeInteger" maxOccurs="1"
minOccurs="1" />
  </xs:sequence>
```

```
</xs:complexType>
<!-- end of RUNRANGE type -->

<!-- define MESSAGE type -->
<xs:complexType name="message_t">
 <xs:sequence>
  <xs:element name="TEXT" type="xs:string" maxOccurs="1" />
 </xs:sequence>
 <xs:attribute name="content" use="required" type="xs:string" fixed="text" />
</xs:complexType>
<!-- end of MESSAGE type -->

<!-- define DEST type -->
<xs:complexType name="dest_t">
 <xs:attribute name="msisdn" use="optional" type="xs:string" />
 <xs:attribute name="contactid" use="optional" type="xs:nonNegativeInteger" />
 <xs:attribute name="groupid" use="optional" type="xs:nonNegativeInteger" />
 <xs:attribute name="bookname" use="optional" type="xs:string" />
</xs:complexType>
<!-- end DEST type -->

<!-- define LIST type -->
<xs:complexType name="list_t">
 <xs:sequence>
  <xs:element name="DEST" minOccurs="1" maxOccurs="unbounded" type="dest_t" />
 </xs:sequence>
</xs:complexType>
<!-- end of LIST type -->

<xs:complexType name="jobnotify_t">
 <xs:sequence>
  <xs:element name="NOTIFYURL" type="xs:string" maxOccurs="1" minOccurs="1" />
 </xs:sequence>
</xs:complexType>

<xs:complexType name="dlrnotify_t">
 <xs:sequence>
  <xs:element name="NOTIFYURL" type="xs:string" maxOccurs="1" minOccurs="1" />
 </xs:sequence>
</xs:complexType>

<!-- define GEOSMS Type -->
<xs:complexType name="geosms_t">
 <xs:sequence>
  <xs:element name="MESSAGE" type="message_t" maxOccurs="1" />
  <xs:element name="LIST" type="list_t" maxOccurs="1" />
```

```
   <xs:element name="TPOA" type="xs:string" maxOccurs="1" minOccurs="0" />
   <xs:element name="DEFERRED" type="xs:string" maxOccurs="1" minOccurs="0" />
   <xs:element name="RUNRANGE" type="runrange_t" maxOccurs="1" minOccurs="0" />
   <xs:element name="JOBNOTIFY" type="jobnotify_t" maxOccurs="1" minOccurs="0" />
   <xs:element name="DLRNOTIFY" type="dlrnotify_t" maxOccurs="1" minOccurs="0" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of GEOSMS Type -->

 <!-- Define completely the message from complex type -->
 <xs:element name="GEOSMS" type="geosms_t" />

</xs:schema>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define smsstatusok_t type -->
 <xs:complexType name="smsstatusok_t">
  <xs:attribute name="MSISDN" use="required" type="xs:string"/>
  <xs:attribute name="ORDERID" use="required" type="xs:string"/>
 </xs:complexType>
 <!-- end of smsstatusok_t type -->

 <!-- define smsstatuserror_t type -->
 <xs:complexType name="smsstatuserror_t">
  <xs:attribute name="MSISDN" use="required" type="xs:string"/>
  <xs:attribute name="REASON" use="required" type="xs:string"/>
 </xs:complexType>
 <!-- end of smsstatuserror_t type -->

 <!-- define sendsmsuser_t type -->
 <xs:complexType name="sendsmsuser_t">
  <xs:attribute name="USERID" use="required" type="xs:string"/>
  <xs:attribute name="SENT" use="required" type="xs:nonNegativeInteger"/>
  <xs:attribute name="CREDIT" use="required" type="xs:decimal"/>
  <xs:attribute name="ACCOUNT" use="required" type="xs:string"/>
  <xs:attribute name="EXPIRE" use="required" type="xs:string"/>
 </xs:complexType>
 <!-- end of sendsmsuser_t type -->

 <!-- define sendsmsreport_t type -->
 <xs:complexType name="sendsmsreport_t">
  <xs:sequence>
   <xs:element name="OK" type="smsstatusok_t"  maxOccurs="unbounded" minOccurs="0"/>
```

```
   <xs:element name="ERROR" type="smsstatuserror_t" maxOccurs="unbounded"
minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="REQUESTED" use="required" type="xs:nonNegativeInteger"/>
  <xs:attribute name="POSTED" use="required" type="xs:nonNegativeInteger"/>
 </xs:complexType>
 <!-- end of sendsmsreport_t type -->

 <!-- define sendsmsbad_t type -->
 <xs:complexType name="sendsmsbad_t">
  <xs:attribute name="CODE" use="required" type="xs:nonNegativeInteger"/>
  <xs:attribute name="DESCRIPTION" use="required" type="xs:string"/>
 </xs:complexType>
 <!-- end of sendsmsbad_t type -->

 <!-- define georesponse_t type -->
 <xs:complexType name="georesponse_t">
  <xs:sequence>
   <xs:element name="USER" type="sendsmsuser_t" maxOccurs="1" minOccurs="0"/>
   <xs:element name="BAD" type="sendsmsbad_t" maxOccurs="1" minOccurs="0"/>
   <xs:element name="REPORT" type="sendsmsreport_t" maxOccurs="1" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
 <!-- end georesponse_t type -->

 <!-- Define completely the message from complex type -->
 <xs:element name="GERESPONSE" type="georesponse_t"/>

</xs:schema>
```

Retrieve SMS status

Request body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define DEST type -->
 <xs:complexType name="geosmsstatus_dest_t">
  <xs:attribute name="msisdn" use="required" type="xs:string" />
  <xs:attribute name="orderid" use="required" type="xs:string" />
 </xs:complexType>
 <!-- end DEST type -->

 <!-- define LIST type -->
 <xs:complexType name="geosmsstatus_list_t">
  <xs:sequence>
```

```
   <xs:element name="DEST" minOccurs="1" maxOccurs="unbounded"
type="geosmsstatus_dest_t" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of LIST type -->

 <!-- define JOB type -->
 <xs:complexType name="geosmsstatus_job_t">
  <xs:attribute name="orderid" use="required" type="xs:string" />
  <xs:attribute name="limit" use="required" type="xs:string" />
 </xs:complexType>
 <!-- end of JOB type -->

 <!-- define RANGE type -->
 <xs:complexType name="geosmsstatus_range_t">
  <xs:attribute name="date_from" use="required" type="xs:string" />
  <xs:attribute name="date_to" use="required" type="xs:string" />
  <xs:attribute name="limit" use="required" type="xs:string" />
 </xs:complexType>
 <!-- end of RANGE type -->

 <!-- define GEOSMSSTATUS Type -->
 <xs:complexType name="geosmsstatus_t">
  <xs:sequence>
   <xs:element name="RANGE" type="geosmsstatus_range_t" maxOccurs="1" minOccurs="0" />
   <xs:element name="LIST" type="geosmsstatus_list_t" maxOccurs="1" minOccurs="0" />
   <xs:element name="JOB" type="geosmsstatus_job_t" maxOccurs="1" minOccurs="0" />
   <xs:element name="ID_CLIENT" type="xs:string" maxOccurs="1" minOccurs="1" />
   <xs:element name="REQUEST_TYPE" type="xs:string" maxOccurs="1" minOccurs="1" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of GEOSMSSTATUS Type -->

 <!-- Define completely the message from complex type -->
 <xs:element name="GEOSMSSTATUS" type="geosmsstatus_t" />

</xs:schema>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define OK/ERROR type -->
 <xs:complexType name="ret_t">
  <xs:attribute name="state" use="required" type="xs:string" />
  <xs:attribute name="msisdn" use="required" type="xs:string" />
```

```
  <xs:attribute name="orderid" use="required" type="xs:string" />
 </xs:complexType>
 <!-- end of OK/ERROR type -->

 <!-- define REPORT type -->
 <xs:complexType name="geosmsstatus_report_t">
  <xs:sequence>
   <xs:element name="OK" type="ret_t" maxOccurs="unbounded" minOccurs="0" />
   <xs:element name="ERROR" type="ret_t" maxOccurs="unbounded" minOccurs="0" />
  </xs:sequence>
  <xs:attribute name="requested" use="required" type="xs:nonNegativeInteger" />
  <xs:attribute name="posted" use="required" type="xs:nonNegativeInteger" />
 </xs:complexType>
 <!-- end of REPORT type -->

 <!-- define GEOSMSSTATUSRESPONSE Type -->
 <xs:complexType name="geosmsstatusresponse_t">
  <xs:sequence>
   <xs:element name="REPORT" type="geosmsstatus_report_t" maxOccurs="1" minOccurs="1" />
   <xs:element name="ID_CLIENT" type="xs:string" maxOccurs="1" minOccurs="1" />
   <xs:element name="REQUEST_TYPE" type="xs:string" maxOccurs="1" minOccurs="1" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of GEOSMSSTATUSRESPONSE Type -->

 <!-- Define completely the message from complex type -->
 <xs:element name="GEOSMSSTATUSRESPONSE" type="geosmsstatusresponse_t" />

</xs:schema>
```

Manage scheduled jobs

Request body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

 <!-- define RUNRANGE type -->
 <xs:complexType name="runrange_t">
  <xs:sequence>
   <xs:element name="L_TIME" type="xs:string" maxOccurs="1" minOccurs="1" />
   <xs:element name="TIME_WINDOW" type="xs:nonNegativeInteger" maxOccurs="1"
minOccurs="1" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of RUNRANGE type -->
```

```
<!-- define dest_jobsmsscheduler_t type -->
<xs:complexType name="dest_jobsmsscheduler_t">
 <xs:attribute name="msisdn" use="optional" type="xs:string" />
 <xs:attribute name="contactid" use="optional" type="xs:nonNegativeInteger" />
 <xs:attribute name="groupid" use="optional" type="xs:nonNegativeInteger" />
 <xs:attribute name="bookname" use="optional" type="xs:string" />
</xs:complexType>
<!-- end dest_jobsmsscheduler_t type -->

<!-- define LISTMSISDN type -->
<xs:complexType name="listmsisdn_t">
 <xs:sequence>
  <xs:element name="ADD" minOccurs="0" maxOccurs="unbounded"
type="dest_jobsmsscheduler_t" />
  <xs:element name="DEL" minOccurs="0" maxOccurs="unbounded"
type="dest_jobsmsscheduler_t" />
 </xs:sequence>
</xs:complexType>
<!-- end of LISTMSISDN type -->

<xs:complexType name="jobnotify_t">
 <xs:sequence>
  <xs:element name="NOTIFYURL" type="xs:string" maxOccurs="1" minOccurs="1" />
 </xs:sequence>
</xs:complexType>

<xs:complexType name="dlrnotify_t">
 <xs:sequence>
  <xs:element name="NOTIFYURL" type="xs:string" maxOccurs="1" minOccurs="1" />
 </xs:sequence>
</xs:complexType>

<!-- define JOB type -->
<xs:complexType name="jobsmsscheduler_t">
 <xs:sequence>
  <xs:element name="TPOA" type="xs:string" maxOccurs="1" minOccurs="0" />
  <xs:element name="MESSAGETEXT" type="xs:string" maxOccurs="1" minOccurs="0" />
  <xs:element name="DEFERREDTIME" type="xs:string" maxOccurs="1" minOccurs="0" />
  <xs:element name="LISTMSISDN" type="listmsisdn_t" maxOccurs="1" minOccurs="0" />
  <xs:element name="RUNRANGE" type="runrange_t" maxOccurs="1" minOccurs="0" />
  <xs:element name="JOBNOTIFY" type="jobnotify_t" maxOccurs="1" minOccurs="0" />
  <xs:element name="DLRNOTIFY" type="jobnotify_t" maxOccurs="1" minOccurs="0" />
 </xs:sequence>
 <xs:attribute name="JOBID" use="required" type="xs:string" />
 <xs:attribute name="ACTION" use="required" type="xs:string" />
</xs:complexType>
<!-- end JOB type -->
```

```
<!-- define GEOSMSSCHEDULER Type -->
<xs:complexType name="geosmsscheduler_t">
 <xs:sequence>
  <xs:element name="JOB" type="jobsmsscheduler_t" maxOccurs="1" minOccurs="1" />
 </xs:sequence>
</xs:complexType>
<!-- end of GEOSMSSCHEDULER Type -->

<!-- Define completely the message from complex type -->
<xs:element name="GEOSMSSCHEDULER" type="geosmsscheduler_t" />

</xs:schema>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<!-- define RUNRANGE type -->
<xs:complexType name="runrange_t">
 <xs:sequence>
  <xs:element name="L_TIME" type="xs:string" maxOccurs="1" minOccurs="1" />
  <xs:element name="TIME_WINDOW" type="xs:nonNegativeInteger" maxOccurs="1"
minOccurs="1" />
 </xs:sequence>
</xs:complexType>
<!-- end of RUNRANGE type -->

<xs:complexType name="jobnotify_t">
 <xs:sequence>
  <xs:element name="NOTIFYURL" type="xs:string" maxOccurs="1" minOccurs="1" />
 </xs:sequence>
</xs:complexType>

<!-- define jobsmsscheduler_t type -->
<xs:complexType name="jobsmsscheduler_t">
 <xs:sequence>
  <xs:element name="TPOA" type="xs:string" maxOccurs="1" minOccurs="0" />
  <xs:element name="MESSAGETEXT" type="xs:string" maxOccurs="1" minOccurs="0" />
  <xs:element name="DEFERREDTIME" type="xs:string" maxOccurs="1" minOccurs="0" />
  <xs:element name="RUNRANGE" type="runrange_t" maxOccurs="1" minOccurs="0" />
  <xs:element name="JOBNOTIFY" type="jobnotify_t" maxOccurs="1" minOccurs="0" />
 </xs:sequence>
 <xs:attribute name="JOBID" use="required" type="xs:string" />
 <xs:attribute name="TRANSACTION" use="required" type="xs:string" />
 <xs:attribute name="CAUSE" use="required" type="xs:string" />
```

```
  <xs:attribute name="CODE" use="required" type="xs:nonNegativeInteger" />
 </xs:complexType>
 <!-- end jobsmsscheduler_t type -->

 <!-- define GEOSMSSCHEDULER Type -->
 <xs:complexType name="geosmsscheduler_t">
  <xs:sequence>
   <xs:element name="JOB" type="jobsmsscheduler_t" maxOccurs="1" minOccurs="1" />
  </xs:sequence>
 </xs:complexType>
 <!-- end of GEOSMSSCHEDULER Type -->

 <!-- Define completely the message from complex type -->
 <xs:element name="GEOSMSSCHEDULER" type="geosmsscheduler_t" />

</xs:schema>
```

Appendix C: related documents

Code	Description