SMS sending using HTTP. XML API

LleidaNetworks Serveis Telematics, S.L.
deval@lleida.net

3rd August 2007
Contents

1 Introduction 4

2 General parameters 4
  2.1 Error codes 4

3 SMS sending format 5
  3.1 Element src 6
  3.2 Element dst 6
  3.3 Element txt 7
  3.4 Element data_coding 8
  3.5 Element delivery_receipt 8
  3.6 Element allow_answer 9
  3.7 Element mt_id 10
  3.8 Element schedule 10
  3.9 Subelements in <result> element for SMS sending operation 11

4 Message status query 12
  4.1 Query format 12
  4.2 Subelements in <result> element for the status query operation 12

5 Incoming new messages query 14
  5.1 Query format 14
  5.2 Element <result> subelements for new incoming messages queries 14

6 User information query 16
  6.1 Query format 16
  6.2 Subelements in <result> element for the user information query operation 16
Copyright

(c) 2007 - LleidaNetworks Serveis Telematics, S.L.

All Rights Reserved

This document is protected by copyright under the International copyright laws and conventions. Without the express written permission of Lleida Networks Serveis Telematics SL, the User may not use, distribute, store, modify, publish, transmit or participate in the transfer, sale or reproduction nor create derivative works from, distribute, perform, display or store any part of its Material for purposes other than the User’s personal, noncommercial use.

User may copy the Material or otherwise download items for personal, noncommercial use, provided the User maintains all copyright information on such Materials and has obtained written permission from Lleida Networks Serveis Telematics, SL.
1 Introduction

The API XML/HTTP allows sending SMS through http queries allowing acceeding to LLeida Networks delivery services from securized networks through firewalls just allowing the outgoing traffic of standard web browsing. Moreover, by using HTTP protocol and XML format for the data representation a quick integration with the client’s application is granted since almost all programming languages have an excellent support for both of them.

2 General parameters

Short messages operation is invoked by the execution of the CGI located in http://sms lleida.net/xmlapi/smsgw.cgi. The request to the CGI should be done with the POST method from HTTP protocol. The request to the CGI should include the variable xml, which will contain the operation data encoded according to the XML format described in later sections.

All the operations receive as reply other XML document, being <result> its root node. This element is made up of several elements common to all the operations, always present and followed by other elements specific for the invoked operation.

The tag <result> common elements are:

- **action**: Id name of the invoked operation
- **status**: Status code of operation success or error
- **msg**: Information text of operation success or error

Apart from these common fields, the <result> element adds other specific elements depending on each operation. These are detailed in their corresponding sections.

2.1 Error codes

The status codes the status parameter admit are:

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Correct</td>
</tr>
<tr>
<td>0</td>
<td>Unknown error</td>
</tr>
</tbody>
</table>
3 SMS sending format

The DTD describing the XML format used for SMS sending can be found at http://sms.lleida.net/xmlapi/docs/send-sms.dtd.

The XML root element must be the tag `<sms>`. This tag should include, at least, the following elements:

- **user**: user’s login in LLeida Networks SMS delivery platform
- **password**: user’s password in LLeida Networks SMS delivery platform
- **dst**: contains one or more `<num>` elements, with the SMS recipient numbers
- **txt**: the SMS text

Next, follows the shortest possible XML format for the sending of a SMS:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<sms>
  <user>mylogin</user>
  <password>mypasswd</password>
  <dst>
    <num>+34600000000</num>
    <num>+34666666666</num>
  </dst>
  <txt>this is the SMS text</txt>
</sms>
```

The following table enumerates all the elements the tag `<sms>` admits:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>XML invalid</td>
</tr>
<tr>
<td>-2</td>
<td>Invalid user</td>
</tr>
<tr>
<td>-3</td>
<td>insufficient credit</td>
</tr>
<tr>
<td>-4</td>
<td>Invalid recipient</td>
</tr>
<tr>
<td>-5</td>
<td>Invalid text</td>
</tr>
<tr>
<td>-6</td>
<td>Temporal error. Retry</td>
</tr>
<tr>
<td>-7</td>
<td>Duplicated session</td>
</tr>
</tbody>
</table>
### 3.1 Element src

This element allows to specify the SMS sender. The sender could be either numerical or alphanumerical. On the first it can not exceed 12 characters, on the second it can not exceed 11 characters.

Example:

```xml
<sms>
  <user>myuser</user>
  <password>mypassword</password>
  <src>mysender</src>
  ...
</sms>
```

### 3.2 Element dst

This element must contain one or more sub-elements `<num>`, each one containing a SMS recipient’s phone number

```xml
<dst>
```
Phone numbers have to come in international format, that is, with the plus sign at the beginning followed by the country code.

There is no specified limit to the quantity of SMS recipients, although it is recommended not to exceed beyond 100.

3.3 Element txt

The `<txt>` element contains the SMS text that is going to be sent. In case the length of the text will be bigger than the limit of characters per SMS allowed (160 in text messages with ASCII encoding and 70 with Unicode), LLeida Networks SMSC will split the message in several concatenated SMS.

This element has two optional attributes, namely `encoding` and `charset`, to be used in case the text of the message is other than the specified in the XML header. For example, in case the XML is encoded in ISO-8859-1 (Latin) and you want to send it in Unicode.

If the SMS text charset is different from the XML one, the encoding parameter must have the "base64" value and the content of the `<txt>` element should be encoded in base 64. Thus, we will avoid the possibility of an invalid XML because of finding characters not allowed in the applied encoding. Moreover, the `charset` parameter must contain the applied characters code (for example, utf-16 for Unicode encoding). When the server finds these parameters, first it decodes base 64 text then it imports it according to the specified charset.

It must be emphasized that `encoding` and `charset` parameters do not affect to the encoding of the resulting SMS: its only function is to include text in any charset in the XML. The SMS data coding, will depend on the content of the element `data_coding` (See section 3.4).

The following example specifies as the text of the SMS the sentence “Hello world”, but using utf-16 charset:

```xml
<txt encoding='base64' charset='utf-16'>
/v8ASAB1AGwAbABvACAAAdwBvAHIAbABk
</txt>
```
3.4 Element data_coding

This optional element controls the data coding of the SMS (see 3GPP specification number TS 23.038). The only two allowed values are: text (default value) and unicode. If the text value is specified, then only GSM alphabet characters (the vast majority of ASCII characters) will be the only ones allowed in the text of the message. On the other hand, with unicode value, worldwide characters can be used (as long as the receiving handset support them) since the SMS is sent in utf-16.

When sending Unicode SMS, you will probably have to combine this element together with the encoding and charset attributes of the <txt> element.

In the following example a SMS with alpha and beta Greek letters is sent:

```xml
<sms>
  <user>myuser</user>
  <password>mypassword</password>
  <dst>
    <num>+3088888888</num>
  </dst>
  <txt encoding='base64' charset='utf-16'>
    /v8DsQ0y
  </txt>
  <data_coding>unicode</data_coding>
</sms>
```

3.5 Element delivery_receipt

This element activates the delivery receipt request for the SMS. This element may be either empty or with an electronic mail address.

With an electronic mail address, an email will be send to the given address whenever the operator notifies any change in the state of the SMS. These notifications are “Delivered”, “Buffered” (switch off or out of network coverage), “acknowledged by the operator SMC” or “Rejected”.

With an empty element, the delivery receipt request is also activated. Once received, the state of the message is stored in LLeida Networks SMSC data base and it can be consulted in the user’s intranet.

Example of a SMS sending with delivery receipt notification via email

```xml
<?xml version="1.0" encoding="utf-8" ?>

<sms>
```
Example of a SMS sending requesting delivery receipt with no electronic mail address.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<sms>
  <user>myuser</user>
  <password>mypassword</password>
  <dst>
    <num>+34600000000</num>
    <num>+34666666666</num>
  </dst>
  <txt>this is the SMS text</txt>
  <delivery_receipt/>
</sms>
```

3.6 Element allow_answer

This element allows to send a message with a long numerical sender thus allowing the addressee to reply to it. Should the user has a allocated number this will be used, otherwise a dynamic one will be allocated Whenever this element is specified the <src> element can not appear. Example:

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>
<sms>
  <user>myuser</user>
```

3.7 Element mt_id

This element allows the user to allocate a unique ID number for each message and subsequently to check the status of the message.

The content of this element has to be a unique alphanumerical string, and the system will add the receiving number to make the id number.

Example:

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>

<sms>
  <user>myuser</user>
  <password>mypassword</password>
  <dst>
    <num>+34600000000</num>
    <num>+34666666666</num>
  </dst>
  <txt>message text</txt>
  <mt_id>ABCDE1234</mt_id>
</sms>
```

The system adds the destination telephone to the id number to administer multisendings. As above mentioned, the generated id number for the first message is “ABCDE1234:+34600000000” and for the second one “ABCDE1234:+34666666666”.

3.8 Element schedule

This element allows to schedule the specific date and time the message should be sent.

Date format should be “YYYYMMDDhhmm”.

Example:
<?xml version="1.0" encoding="iso-8859-1" ?>

<sms>
  <user>myuser</user>
  <password>mypassword</password>
  <dst>
    <num>+34600000000</num>
    <num>+34666666666</num>
  </dst>
  <txt>messagetext</txt>
  <schedule>200712011350</schedule>
</sms>

In this example the text message is scheduled to be sent to both addressees on 1st December 2007 at 15.30.

3.9 Subelements in <result> element for SMS sending operation

When a sms sending ends up successfully the element has, apart from the three common elements, the <newcredit> one. This sub-element informs of the available credit the user has once the sms has been sent.
4 Message status query

By means of this command you may consult the status of a MT. However, this option is only possible when the sending was carried out specifying an id number through the tag \texttt{mt\_id}.

4.1 Query format

The \textit{root element} in the XML document for the status query is \textit{root element} \texttt{query\_mt\_status}. Within this element the following fields are mandatory:

- \texttt{user}: User login in Lleida.net SMS delivery platform
- \texttt{password}: User password in Lleida.net SMS delivery platform
- \texttt{mt\_id}: Message ID number the user supplies in the sending

Example:

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>
<query_mt_status>
  <user>myuser</user>
  <password>mypassword</password>
  <mt_id>BD4fSA56:+3460000000</mt_id>
</query_mt_status>
```

4.2 Subelements in \texttt{<result>} element for the status query operation

The \textit{root element} from the result of a MT status query is the \texttt{<result>} element, containing the common subelements (see section 2 and the \texttt{<mt\_status>} element.

The \texttt{<mt\_status>} has the following subelements:

- \texttt{mt\_id}: Message id number from which status has been asked
- \texttt{status\_code}: Message status code.
- \texttt{status\_desc}: Descriptive message of the message status.
- \texttt{tm\_last\_update}: Unix timestamp on the status message since the last modification
Example:

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>

<result>
  <action>query_mt_status</action>
  <status>100</status>
  <msg>Success</msg>
  <mt_status>
    <mt_id>BD4fSA56:+3460000000</mt_id>
    <status_code>3</status_code>
    <status_desc>Sent</status_desc>
    <tm_last_update>1148990517</tm_last_update>
  </mt_status>
</result>
```

The possible status codes are:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New</td>
<td>The message has not been processed yet</td>
</tr>
<tr>
<td>2</td>
<td>Pending</td>
<td>the message has not been sent yet</td>
</tr>
<tr>
<td>3</td>
<td>Sent</td>
<td>The message has been sent.</td>
</tr>
<tr>
<td>4</td>
<td>Delivered</td>
<td>message has been delivered to the addressee(mobile phone confirmation has been received).This status is only available if delivery receipt was activated in the sending</td>
</tr>
<tr>
<td>5</td>
<td>Buffered</td>
<td>Message has been sent to operator yet not to addresse .Mobile switched off or out of range. Operator will retry sending until recipient received sms This status is only available if delivery receipt was activated in the sending.</td>
</tr>
<tr>
<td>6</td>
<td>Failed</td>
<td>The message has not been sent</td>
</tr>
</tbody>
</table>
5 Incoming new messages query

LleidaNetworks delivery platform allows both the sending and reception of messages. The polling pattern is being used to receive messages where the client asks the system on a regular basis and the system in its turn reply with the new messages since the last query.

5.1 Query format

The XMLThe root element in the XML document for the status query is tag \texttt{get\_new\_incoming\_mo}. Within this element the following fields are mandatory:

- \texttt{user}: User login in Lleida.net SMS delivery platform
- \texttt{password}: User password in Lleida.net SMS delivery platform

5.2 Element \texttt{result} subelements for new incoming messages queries

The root element of the new incoming messages query result is the \texttt{result} which includes the common elements (see section 2) and the \texttt{mo\_list} element. The \texttt{mo\_list} element contains a list of all the new available messages each of them encapsulated within a \texttt{mo} element. The \texttt{mo} element contains the following subelements:

- \texttt{id}: Incoming message ID.
- \texttt{tm\_rec}: Unix timestamp message date receipt.
- \texttt{src}: Message sender number.
- \texttt{dst}: Message addressee number.
- \texttt{txt}: Message full text.

Example:

```xml
<?xml version="1.0" encoding="iso-8859-1" ?>

<result>
  <action>get_new_incoming_mo</action>
  <status>100</status>
</result>
```
<msg>Success</msg>
<mo_list>
  <mo>
    <id/fa439d93c2</id>
    <tm_rec>1148990517</tm_rec>
    <src>+34666666666</src>
    <dst>+34973900101</dst>
    <txt>Text message 1</txt>
  </mo>
  <mo>
    <id/fa439d93c3</id>
    <tm_rec>1148990520</tm_rec>
    <src>+34666666666</src>
    <dst>+34973900101</dst>
    <txt>Text message 2</txt>
  </mo>
</mo_list>
</result>
6 User information query

By means of this command you may consult your user information.

6.1 Query format

The root element in the XML document for the user information query is the tag \texttt{userinfo}. Within this element the following fields are mandatory:

- \texttt{user}: User login in Lleida.net SMS delivery platform
- \texttt{password}: User password in Lleida.net SMS delivery platform

6.2 Subelements in <result> element for the user information query operation

The root element from the result of a user information query is the <result> element, containing the common subelements (see section 2) and the <userinfo> element.

Example:

```xml
<result>
  <action>userinfo</action>
  <status>100</status>
  <msg>Success</msg>
  <userinfo>
    <user>User login</user>
    <credit>User credit</credit>
    <status>User status</status>
    <created>Registered date timestamp</created>
    <lastop>Last operation timestamp</lastop>
    <contact_name>Contact person</contact_name>
    <phone>Phone number</phone>
    <email>Contact email</email>
    <organization>Organization name</organization>
    <cif>Organization ID</cif>
    <phone_number>
      <num>User allocated number</num>
    </phone_number>
  </userinfo>
</result>
```