myMSP API v2 Appendix HTTPS

System Interface - Specifications

1 Purpose of Document

The purpose of this document is to define the HTTPS interface made available by myMSP to access the application functionality. Accessing myMSP can be done from the computer systems of external parties/customers.

1.1 Author

21st Century Mobile
Mikael Rosvall
mikael.rosvall@21st.se
+46 (0)8 21 21 55

1.2 Document Status

- Published 2007-01-08

1.2.1 Document Changes

- B [2007-06-15]
  - Minor changes and improvements
- C [2007-12-11]
  - New method added: Retrieve messages
    - 3.3: Receive a message
    - 4.3: Retrieve received messages
- D [2008-03-28]
  - Updated all URL-references from the old domain (www.mymsp.eu) to the new domain (mymsp.21st.se).
  - Changed the description of the recipients-field in chapter 4.1.1 to include recipient groups.
- E [2009-04-29]
  - New parameter added: delivery time (4.1.1 and 4.1.3)
- F [2009-12-07]
  - New parameter added: charset (4.1.1)
- G [2010-12-17]
  - New parameter added: isflash (4.1.1)
  - New parameter added: destinationport (4.1.1)
- H [2011-08-24]
  - New error code added: 12 (4.1.3)
- I [2012-10-26]
  - Updated error codes (4.1.3, 4.2.3, 4.4.3)
- J [2015-01-28]
  - New methods added: Delivery time (4.3)
- K [2015-09-09]
  - Added “GET” to the supported HTTP methods in the Technical description (2)
- L [2018-11-02]
  - Removed error code 301.
- M [2019-03-11]
  - Updated graphic profile.
  - New methods added: Retrieve uncollected delivery reports and received messages (4.5)

1.3 Document Perimeter

Neither economical questions nor questions regarding agreements/contracts will be dealt with in this document.

The information given in this document may change without notice and describes only the matters defined in the general part of this document. Please verify that your company has the most recent version. This information is intended for the use of customers and partners of 21st Century Mobile. The information or statements given in this document concerning the suitability, capacity or performance of the mentioned service cannot be considered binding but shall be defined in the agreement concluded between 21st Century Mobile and the customer, if applicable. 21st Century Mobile shall not be responsible in any event for errors in this document or for any damages, incidental or consequential (including monetary losses), that might arise from the use of this publication or the information in it. This material and the service described in this document are copyrighted in accordance with the applicable laws.

2 Technical description

This interface is realised through HTTPS. An external system can send data to myMSP through HTTPS using the GET or POST methods to submit data. The results are delivered as a HTTPS RESPONSE in the “text/plain” format. Each function is accessed through a specific URL, i.e. to send a SMS the following URL is used:

- https://mymsp.21st.se/external/sendSms

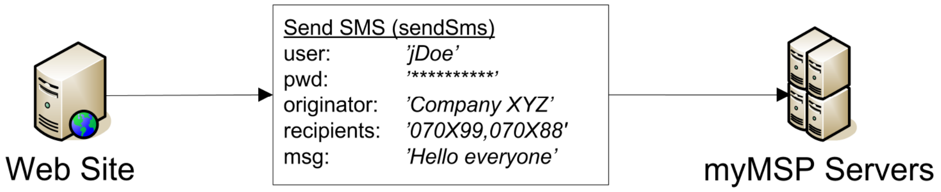
3 Usage scenario

The following scenarios are intended to give a brief overview of how to use the myMSP HTTPS interface. In the next chapter the API is described in more detail.

3.1 Send a message

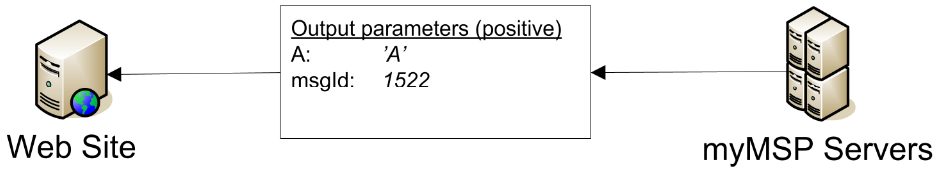
As previously mentioned in the main API-document (3.1 The send message sequence) sending a message through myMSP is done in steps.

1. First, a message *send request* is sent to myMSP. The request contains user identification (user name and password), a string of the recipients' numbers, and the message content.



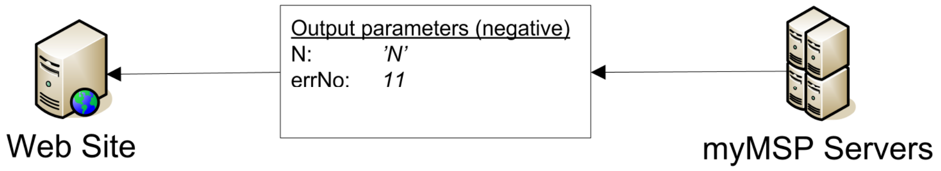
Send SMS (sendSms)
user: *'jDoe'*
pwd: *'**********'*
originator: *'Company XYZ'*
recipients: *'070X99,070X88'*
msg: *'Hello everyone'*

2. After the send request has been sent, a set of parameters is returned.
   a. If myMSP has accepted the request then a set of positive parameters is returned, including the ID that myMSP has assigned the message.



Output parameters (positive)
A: *'A'*
msgId: *1522*

   b. If myMSP has rejected the request (e.g. the password for the user was incorrect) then a set of negative parameters are returned, including the specific code of the error.

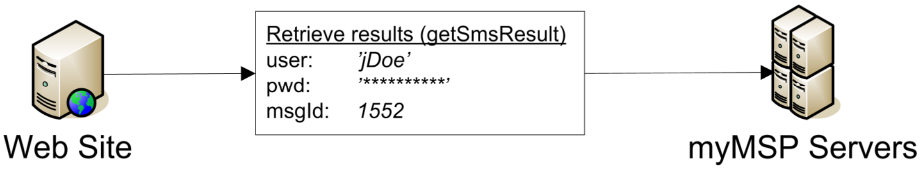


Output parameters (negative)
N: *'N'*
errNo: *11*

3.2 Check the status of a sent message

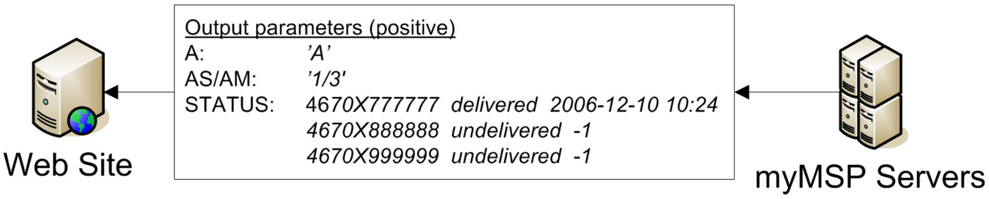
It can take some time before a message is delivered to all recipients. The status of a message can be checked while it is being delivered.

1. To check the current status of a message a *retrieve result request* has to be sent to myMSP. The request is composed of user identification, and the message number of the message whose status should be checked.



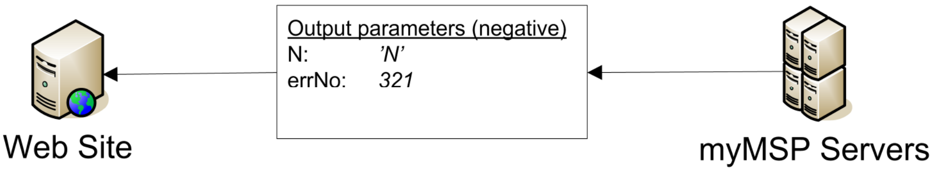
Retrieve results (getSmsResult)
user: *’jDoe’*
pwd: ’**********’
msgId: *1552*

2. After the retrieve result request has been sent, a set of parameters is returned.
   a. If myMSP has accepted the request then a set of positive parameters is returned, including information regarding the send status for each recipient, e.g. if the message has been successfully sent to a specific recipient.



Output parameters (positive)
A: *’A’*
AS/AM: *’1/3’*
STATUS: *4670X777777 delivered 2006-12-10 10:24*
4670X888888 undelivered -1
4670X999999 undelivered -1

   b. If myMSP has rejected the request then a set of negative parameters are returned, including the specific code of the error.



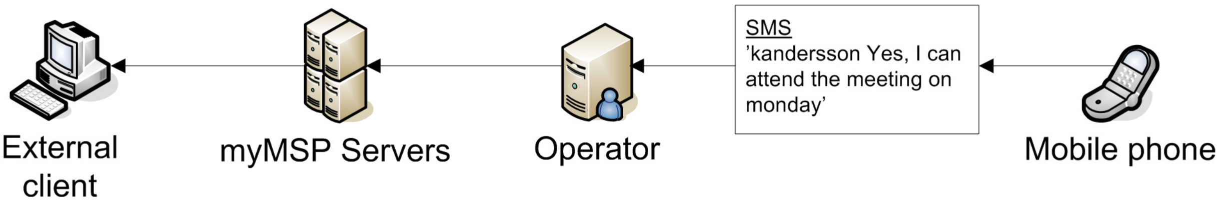
Output parameters (negative)
N: *’N’*
errNo: *321*

The process checking the send status of a message can be repeated as many times as is desirable, e.g. to check and make sure that all recipients have received the message.

3.3 Receive a message

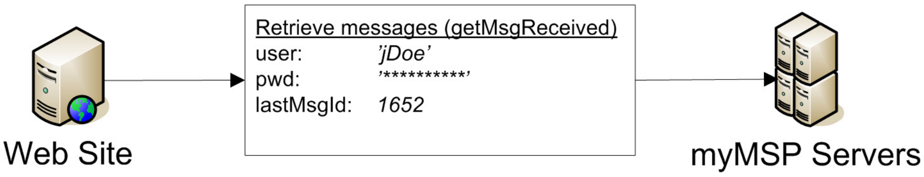
myMSP can also receive messages from mobile phones.

1. An initial ID is needed in order to send a message to a myMSP account from a mobile phone. The initial ID is placed in the beginning of the text in the message and should be followed by a blank space.



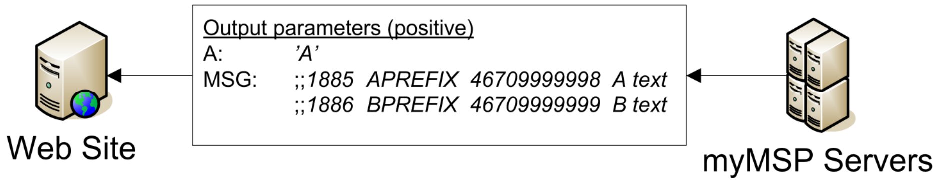
SMS
’kandersson Yes, I can attend the meeting on monday’

2. A *received message request* is needed in order to get new messages that have been sent to a myMSP account.



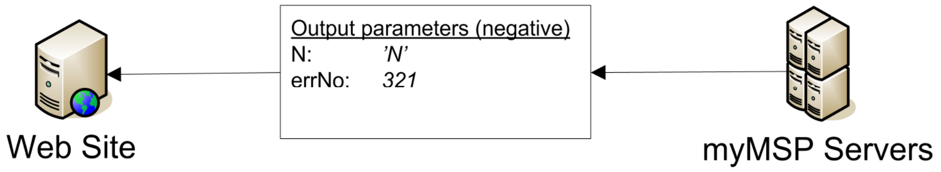
Retrieve messages (getMsgReceived)
user: *’jDoe’*
pwd: ’**********’
lastMsgId: *1652*

3. After the received message request has been sent, a set of parameters is returned.
    a. If myMSP has accepted the request then a set of positive parameters is returned, including the received messages.



Output parameters (positive)
A: *'A'*
MSG: ;;*1885 APREFIX 46709999998 A text*
;;*1886 BPREFIX 46709999999 B text*

    b. If myMSP has rejected the request then a set of negative parameters are returned, including the specific code of the error.



Output parameters (negative)
N: *'N'*
errNo: *321*

4 HTTPS methods

This chapter describes the various HTTPS methods in more detail.

4.1 Send SMS (sendSms)

Send a message to recipients. Note that the web page character encoding, e.g. “charset=UTF-8”, is used to set the character set of the message.

Connection: https://mymsp.21st.se/external/sendSms

4.1.1 Input parameters

user	User Account
pwd	Password
originator	Originator text of the message (see ‘Originator text’ in main API-document glossary). Note that only originator texts that have been pre-registered with myMSP can be used.
recipients	Phone Number of recipients or name/short name of a recipient group created in the myMSP web GUI (https://mymsp.21st.se/app). Should be in the form of a comma-separated string, e.g. “46709888888,46702111111,recipgroupA”.
msg	The message in plain text. Maximum of 765 characters.
deliverytime	The date and time that the message should be delivered. Should be formatted as follows: YYYYMMDDhhmmzz, e.g. 200904291230+1. The last field (zz) is the time zone difference in hours from Western European Time (UTC+0) i.e. GMT. +0 or -0 equals GMT, +1 equals Central European Time (UTC+1), etc.
charset	The character set that was used to encode the message text. If this parameter is not set then ‘ISO-8859-1’ used to decode the message.
isflash	Set to “true” if the message should be sent as a flash SMS.
destinationport	If set and larger then zero then the message is sent to the specified port on the receiver’s mobile phone.

4.1.2 Output parameters (positive result)

A	Character ‘A’
msgId	Message identification number in myMSP. Used when querying send results.

4.1.3 Output parameters (negative result)

N	Character ‘N’
errNo	The error code: 7: Authorization missing 11: Maximum number of messages exceeded 12: Not enough credits left to send message 13: Unknown error, contact support 22: No value in parameter ‘user’ 23: Parameter ‘user’ missing

	24: No value in parameter ‘pwd’ 25: Parameter ‘pwd’ missing 28: One or more of the recipient numbers are in the wrong format 282: No value in parameter ‘recipients’ 29: Parameter ‘recipients’ missing 30: No value in parameter ‘msg’ 31: Parameter ‘msg’ missing 33: Sender alias is not registered 34: Invalid format for delivery time 35: Invalid date for delivery time (must be after current time) 36: Invalid flash value 37: Invalid destination port value 38: Invalid express value 303: Invalid UCP message

4.2 Retrieve send results (getSmsResult)

Check the send result of a message.

Connection: https://mymsp.21st.se/external/getSmsResult

4.2.1 Input parameters

user	User Account
pwd	Password
msgId	Message identification number in myMSP.

4.2.2 Output parameters (positive result)

A	Character ‘A’
R/S/D	The number of recipients (R), sent messages (S) and delivered messages (D). For example ‘10/10/9’ means that a message with ten recipients has been sent to as ten SMS (one to each recipient) and that nine of the messages have been delivered.
STATUS	The delivery status for each recipient of the message is listed. The different fields of a delivery status are separated by a tab-character ( ). The fields are: *Recipient Number;* *Sent time (YYYY-MM-DD hh:mm);* *Delivery status;* *Delivery time (YYYY-MM-DD hh:mm);* If the message has still not been sent to a recipient, minus one (-1) is shown instead of the sent time. If the message has still not been delivered to a recipient, minus one (-1) is shown instead of the delivered time. Example: 4670X666666 2007-12-02 15:14 delivered 2007-12-02 15:15

	4670X777777 2007-12-02 15:15 delivered 2007-12-02 15:18 4670X888888 2007-12-02 15:15 undelivered -1 4670X999999 -1 undelivered -1

4.2.3 Output parameters (negative result)

N	Character ‘N’
errNo	The error code: 7: Authorization missing 13: Unknown error, contact support 22: No value in parameter ‘user’ 23: Parameter ‘user’ missing 24: No value in parameter ‘pwd’ 25: Parameter ‘pwd’ missing 32: Can’t find message with msgId 323: Message has been archived 391: Parameter ‘msgId’ is invalid 392: No value in parameter ‘msgId’

4.3 Delivery time

4.3.1 Get delivery time (sendlater/get)

Gets the delivery time of a message that has been sent to myMSP.

Connection https://mymsp.21st.se/external/sendlater/get

4.3.1.1 Input parameters

user	User Account
pwd	Password
msgno	Message identification number in myMSP.

4.3.1.2 Output parameters (positive result)

A	Character ‘A’
STATUS	The delivery status of the message. The fields are separated by a tab-character ( ). The fields are: *Message ID;* *Delivery time (YYMMDDhhmmssZZZZ);* If the message does not have a delivery time an empty string is returned in the delivery time field. Example: 12556 150112153000004+

4.3.1.3 Output parameters (negative result)

N	Character 'N'
errNo	The error code: 7: Authorization missing 13: Unknown error, contact support 22: No value in parameter 'user' 23: Parameter 'user' missing 24: No value in parameter 'pwd' 25: Parameter 'pwd' missing 32: Can't find message with msgno 391: Parameter 'msgno' is invalid 392: No value in parameter 'msgno'

4.3.2 Get the delivery time for all scheduled messages (sendlater/getall)

Gets the delivery times of all scheduled messages that have not yet been sent by myMSP.

Connection https://mymsp.21st.se/external/sendlater/getall

4.3.2.1 Input parameters

user	User Account
pwd	Password

4.3.2.2 Output parameters (positive result)

A	Character 'A'
STATUS	The delivery status of all scheduled messages that have not yet been sent. The fields are separated by a tab-character ( ). The fields are: *Message ID;* *Delivery time (YYMMDDhhmmssZZZZ);* Example: 12556 150112153000004+ 12557 150115094500004+

4.3.3 Output parameters (negative result)

N	Character 'N'
errNo	The error code: 7: Authorization missing 13: Unknown error, contact support 22: No value in parameter 'user' 23: Parameter 'user' missing 24: No value in parameter 'pwd' 25: Parameter 'pwd' missing

4.3.4 Update delivery time (sendlater/update)

Updates the delivery time of a scheduled messages that has not yet been sent by myMSP.

Connection https://mymsp.21st.se/external/sendlater/update

4.3.4.1 Input parameters

user	User Account
pwd	Password
msgno	Message identification number in myMSP.
deliverytime	The date and time that the message should be delivered. See 4.1.1 for more details.

4.3.4.2 Output parameters (positive result)

A	Character 'A'

4.3.4.3 Output parameters (negative result)

N	Character 'N'
errNo	The error code: 7: Authorization missing 13: Unknown error, contact support 22: No value in parameter 'user' 23: Parameter 'user' missing 24: No value in parameter 'pwd' 25: Parameter 'pwd' missing 32: Can't find message with msgno 34: Invalid format for delivery time 35: Invalid date for delivery time (must be after current time) 351: Can't update delivery time, message already sent 391: Parameter 'msgno' is invalid 392: No value in parameter 'msgno'

4.3.5 Cancel scheduled message (sendlater/cancel)

Cancels a scheduled message.

Connection https://mymsp.21st.se/external/sendlater/cancel

4.3.5.1 Input parameters

user	User Account
pwd	Password
msgno	Message identification number in myMSP.

4.3.5.2 Output parameters (positive result)

A	Character 'A'

4.3.5.3 Output parameters (negative result)

N	Character 'N'
errNo	The error code: 7: Authorization missing 13: Unknown error, contact support 22: No value in parameter 'user' 23: Parameter 'user' missing 24: No value in parameter 'pwd' 25: Parameter 'pwd' missing 32: Can't find message with msgno 351: Can't cancel message, message already sent 391: Parameter 'msgno' is invalid 392: No value in parameter 'msgno'

4.4 Retrieve received messages (getMsgReceived)

Fetch received messages.

Connection: https://mymsp.21st.se/external/getMsgReceived

4.4.1 Input parameters

user	User Account
pwd	Password
lastMsgId	Message identification number in myMSP. All messages with a higher message ID (i.e. messages received later) are returned. Input 0 to fetch all received messages.
clean	Set to "true" to wash all return values, i.e. all new line characters (\r\n, \n, \r) are replaced with the string "\n" and all tab characters (\t) with a space.

4.4.2 Output parameters (positive result)

A	Character 'A'

STATUS	All received message with a msgID-value higher than the one supplied in the request are listed.
	Two semicolon (;;) are placed in front of each message. The different fields of a received message are separated by a tab-character ( ).
	The fields are:
	Message ID;
	Received date (YYYY-MM-DD hh:mm);
	User name;
	Prefix used/Device notification action;
	Message type (SMS/VOI/DPN);
	Originator number;
	Originator alias/Recipient ID (if DPN);
	MMS-subject; (deprecated)
	Message text

	Example: ;;12556 2007-12-02 15:14 jDoe APREFIX SMS 46769999999 null null A text ;;12554 2007-08-19 12:52 jDoe APREFIX SMS 46769999998 null null B text

4.4.3 Output parameters (negative result)

N	Character 'N'
errNo	The error code: 7: Authorization missing 13: Unknown error 22: No value in parameter 'user' 23: Parameter 'user' missing 24: No value in parameter 'pwd' 25: Parameter 'pwd' missing 391: Invalid 'msgno' value 392: Parameter 'msgno' missing

4.5 Retrieve uncollected delivery reports and received messages (getMsgUpdates)

Gets all uncollected delivery reports and received messages for an account. Subsequent calls to this method only returns updates that have happened since the last call. Contact support to activate this function.

Connection: https://mymsp.21st.se/external/getMsgUpdates

4.5.1 Input parameters

user	User Account
pwd	Password

4.5.2 Output parameters (positive result)

A	Character 'A'

STATUS	All uncollected delivery reports and received messages are listed.
	
	Two semicolon (;;) are placed in front of each delivery report/message. The different fields are separated by a tab-character ( ).
	
	The fields for delivery reports are:
	Type (value "0" for delivery reports);
	Message ID;
	Recipient ID;
	Mobile no;
	Delivery status;
	Delivery time (YYMMDDhhmmssZZZZ);
	
	The fields for received messages are:
	Type (value "1" for received messages);
	Message ID;

	Originator number;
	Received date (YYYY-MM-DD hh:mm);
	Prefix used/Device notification action;
	Message type (SMS/VOI/DPN);
	Message text
	
	Example:
	;;0 12556 98765 46769999999 delivered 2007-12-02 15:14
	;;1 12554 46769999998 2019-03-01 12:52 APREFIX SMS B text

4.5.3 Output parameters (negative result)

N	Character ‘N’
errNo	The error code: 7: Authorization missing 13: Unknown error 22: No value in parameter ‘user’ 23: Parameter ‘user’ missing 24: No value in parameter ‘pwd’ 25: Parameter ‘pwd’ missing

5 Examples

Here are some code examples that illustrate how to use the HTTPS API to communicate with myMSP.

5.1 ASP

The results are delivered as “HTTPS RESPONSE” in the ”text/plain” format directly after myMSP has received the message (and all recipients). In other words, the results are shown in text with information on how the message was handled. In order to process this information, for example save it to a database, some form of written code is necessary.

```
<%
'Uses WinHTTP v5.1 from Microsoft
'http://msdn.microsoft.com/library/en-us/winhttp/http/winhttp_start_page.asp

Dim objWinHttp
Dim strHTML
Dim strReq

strReq = "user=" & "myUsername" & _
        "&pwd=" & "myPWD" & _
        "&originator=" & "mySignature" & _
        "&recipients=" & "recipient phone numbers" & _
        "&msg=" & "message text"

Set objWinHttp = Server.CreateObject("WinHttp.WinHttpRequest.5.1")
objWinHttp.Open "POST", " https://mymsp.21st.se/external/sendSms"

objWinHttp.SetRequestHeader "Content-Type", "application/x-www-form-urlencoded"

objWinHttp.Send strReq

'gets/scans results
strHTML = objWinHttp.ResponseText
Set objWinHttp = Nothing
%>

<%
'prints results
= strHTML
%>
```

5.2 HTML

5.2.1 Send SMS

The results are delivered as “HTTPS RESPONSE” in the ”text/plain” format directly after myMSP has received the message (and all recipients). In other words, the results are shown in the browser used to send the SMS with information on how the message was handled. In order to process this information, for example save it to a database, some form of written code is necessary.

```
<form action=" https://mymsp.21st.se/external/sendSms" method="post">
<TABLE cellSpacing="0" cellPadding="5" width="300" border="0">
<TR>
<TD vAlign="middle" noWrap align="right">Username</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="text" name="user"></TD>
<TD><FONT color="#ff0066">*</FONT></TD>
</TR>
<TR>
<TD vAlign="middle" noWrap align="right">Password</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="password" name="pwd"></TD>
<TD><FONT color="#ff0066">*</FONT></TD>
</TR>
<TR>
<TD vAlign="middle" noWrap align="right">Sender</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="text" name="originator"></TD>
<TD></TD>
</TR>
<TR>
<TD vAlign="middle" noWrap align="right">Recipient</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="text" name="recipients"></TD>
<TD><FONT color="#ff0066">*</FONT></TD>
</TR>
<TR>
<TD vAlign="middle" noWrap align="right">Text</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="text" name="msg"></TD>
<TD><FONT color="#ff0066">*</FONT></TD>
</TR>
<TR>
<TD vAlign="middle" noWrap align="left" colSpan="3">
<INPUT type="submit" value="Submit"></TD>
</TR>
</TABLE>
</form>
```

5.2.2 Retrieve SMS

The results are delivered as “HTTPS RESPONSE” in the ”text/plain” format. When sending messages *en masse* there will be a short delay. In other words, the result contains information on the send progress of the messages, which is displayed in text. In order to process this information, for example save it to a database, some form of written code is needed on your part

```
<form action=" https://mymsp.21st.se/external/getSmsResult" method="post">
<TABLE cellSpacing="0" cellPadding="5" width="300" border="0">
<TR>
<TD vAlign="middle" noWrap align="right">Username</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="text" name="user"></TD>
<TD><FONT color="#ff0066">*</FONT></TD>
</TR>
<TR>
```

```
<TD vAlign="middle" noWrap align="right">Password</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="password" name="pwd"></TD>
<TD><FONT color="#ff0066">*</FONT></TD>
</TR>
<TR>
<TD vAlign="middle" noWrap align="right">Message Identification Number</TD>
<TD vAlign="middle" noWrap align="left">
<INPUT type="text" name="msgId"></TD>
<TD><FONT color="#ff0066">*</FONT></TD>
</TR>
<TR>
<TD vAlign="middle" noWrap align="left" colSpan="3">
<INPUT type="submit" value="Submit"></TD>
</TR>
</TABLE>
</form>
```