wireless services

SMS Outbound Premium

HTTP interface - v1.1

Table of contents

- This Page Intentionally Left Blank -

1. Version history

Date	Changes
October 16, 2006	Initial release
November 8, 2006	Parameter DATEFORMAT added and DATE description updated

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

3. Introduction

This document is intended for developers who are responsible for the implementation of the HTTP connection with the SMS/MMS gateway of Wireless Services.

In chapter 4 a description is given of the Application Programming Interface (API) which is available for several programming languages. Chapter 5 contains the necessary information for connecting to the gateway. All available HTTP request parameters are discussed in detail in chapter 6. The gateway response is explained in chapter 7, followed by an example HTTP request and response in chapter 8.

4. Application Programming Interface (API)

Wireless Services has developed an Application Programming Interface (API) which can be easily integrated into other applications. Clients are strongly encouraged to use this API when implementing the HTTP interface for a supported programming language (see table 4.1).

The advantages of using the API are:

Easy integration into client applications
Highly reliable due to redundant gateways
New features will be added when available
Easy upgradable
Outbound and Inbound functionality
IP-based security checks for inbound messages
SHA1 encryption for authentication

The currently supported API functionalities are:

SMS Outbound
- Text SMS
- Premium SMS
- Binary SMS
- Picture SMS
- EMS
- Monophonic ringtones
- WAP Push
- OMA Client Provisioning
- vCalendar
- vCard

SMS Inbound
SMS Delivery Notifications
MMS Outbound

Clients can easily upgrade to the latest version as newer versions are backwards compatible. The API can be downloaded from the Support Center on our website and is available for the programming languages listed in the table below.

Programming language	Current version
Perl 5	2.4
PHP 4 + PHP 5	2.42

Table 4.1 – API programming languages

Please refer to the documentation and examples included with the API for more information.

5. Gateway connection

Clients can connect to the SMS/MMS gateway using either HTTP or HTTPS. Both GET and POST requests are supported, but POST requests are recommended because of limitations in the length of GET requests. The connection details for using the interface described in this document are listed in the table 5.1. The port number for connecting to the gateway is port 80 for HTTP and port 443 for HTTPS.

Gateway	URL	HTTPS
Main (preferred)	http://gateway.wireless-services.nl/sendpsms	Yes
Backup	http://gateway2.wireless-services.nl/sendpsms	No

Table 5.1 – Connection details for HTTP

5.1 Main and backup gateway

The main gateway is the recommended gateway to use for all clients. The backup gateway should be used as a failover server. It is strongly advised to implement application logic to automatically switch to the backup gateway in case the main gateway is unreachable due to Internet connectivity issues.

5.2 Content-Type header

In case of POST requests, the Content-Type header MUST have the following value:

application/x-www-form-urlencoded

5.3 Host header

Clients MUST use the Host header in all HTTP requests to the gateway. The Host header specifies the Internet host of the resource being requested and equals the host portion of the URL (e.g. *gateway.wireless-services.nl* for the main gateway). Please refer to chapter 8 for an example request with Host header.

6. Request parameters

In this chapter all available request parameters will be discussed. The parameter names are case-sensitive. They MUST be written in uppercase letters and all values MUST be properly URL escaped.

The following parameters are defined:

Parameter	Mandatory	Description	Paragraph
VERSION	Yes	Interface version	6.1
UID	Yes	Username of the client's account	6.2
PWD	Yes	Password	6.3
AUTHTYPE	No	Authentication type	6.4
AUTHTIME	No	Authentication timestamp	6.5
N	Yes	Recipient numbers	6.6
M	Yes	Message body	6.7
RATE	Yes	End-user price	6.8
OPR	No	Recipient operator code	6.9
SESSIONID	No	Session identifier	6.10
O	No	Originator	6.11
ONUM	No	Numeric originator	6.12
TEST	No	Test request	6.13
DATE	No	Scheduled delivery	6.14
DATEFORMAT	No	Scheduled delivery date format	6.15

NOT	No	Notification request	6.16
NOTTYPE	No	Notification type	6.17
BATCHID	No	Batch identifier	6.18
MSGID	No	Message identifier for logging purposes	6.19
VAL	No	Validity	6.20

Table 6.1 – HTTP request parameters

Each parameter is described in the next paragraphs.

6.1 Version (VERSION)

The version parameter specifies the interface version the client is using. The current version number described in this document is **1.1**. Clients MUST specify this version number as the value of the parameter.

Default value	1.0
Value range	1.0 – 1.1

Parameter properties

6.2 Username (UID)

To authenticate the client, the username of the client's account must be specified. The account details are sent to the client by our Support department.

It is also possible to use a predefined whitelist of IP numbers/ranges from which the client is allowed to connect. By default, the client can connect from any IP number. The IP whitelist can be enabled by contacting our Support department.

6.3 Password (PWD)

To authenticate the client a password must be supplied. The value of this parameter depends on the used encryption mechanism (see 6.4).

6.4 Authentication type (AUTHTYPE)

The client can choose several encryption mechanisms for authentication. When ***plaintext*** (default) is used, the client's password is transmitted as normal text in the PWD parameter.

To provide a higher level of security and prevent the password from being visible in the request, the encryption mechanisms ***md5*** and ***sha1*** can be used. The input for these encryption functions is the client's account password and the current timestamp, *AUTHTIME* (see 6.5). The output is a 32-character hexadecimal string for MD5 or a 40-character hexadecimal string for SHA1. The computation of the *PWD* parameter is as follows:

PWD = MD5(<password><timestamp>)
PWD = SHA1(<password><timestamp>)

Example:
password=as4bY3
timestamp=1160989330

MD5=7b04fa4523a238b89af4ad63acaa3b00
SHA1=b0d94617b5c49c0284e5983cbfb4355ac64eb654

Many programming languages have built-in functions for MD5 and SHA1 encryption. For more information about these encryption mechanisms, please see RFC1321 (MD5) and RFC3174 (SHA1).

Default value	plaintext
Value range	plaintext / md5 / sha1

Parameter properties

6.5 Authentication timestamp (AUTHTIME)

When the client uses MD5 or SHA1 authentication, the gateway needs to verify the calculated value in the *PWD* parameter with the client's password stored in the database. Therefore, the value of the timestamp used by the client to calculate this

value MUST be transmitted in the *AUTHTIME* parameter. The gateway will perform the same calculation as the client and rejects the request if the results do not match.

The timestamp is specified as a UNIX timestamp, which is the number of elapsed seconds since January 1st, 1970 00:00:00. It is important to note that the gateway will reject requests if the timestamp provided is more than 12 hours earlier or later than the current timestamp. This means that if the internal clock of the client's server is off by more than 12 hours, the request will be rejected.

Default value	None
Value range	12 hours from current time

Parameter properties

6.6 Recipient numbers (N)

The recipient parameter is a comma-separated list of mobile numbers to receive the message. The mobile numbers should be in international format (e.g. +447123456789) with or without leading + sign. The only exceptions to this rule are Dutch mobile numbers. These can also be specified in national format (e.g. 0612345678) and are automatically converted to international format by the gateway.

6.7 Message body (M)

This parameter contains the body of the message. The maximum length for text messages is 160 characters. Longer messages will be truncated.

6.8 End-user price (RATE)

The end-user price of the message is specified in cents. So if the client wishes to send a message with a tariff of EUR 0.70, the value of this parameter must be set to **70**. Contact your account manager for a list of available/enabled end-user prices. If the client specifies a value of **0** (default), the message will not be charged to the end-user.

Default value	0
Value range	0 – 500

Parameter properties

6.9 Operator code (OPR)

For Premium SMS outside the Netherlands the client MUST specify the destination operator code for the specified recipients. The defined operator codes are supplied by Wireless Services when Premium SMS is activated for your account.

Please note that when the client specifies an incorrect or invalid operator code the end-users cannot be charged and the client will not receive the kickback for these messages.

6.10 Session identifier (SESSIONID)

Some operators include a session identifier with each Mobile Originated (MO) message. When the client is sending back a Premium SMS message as a response to an MO message which includes this parameter, they MUST specify the same SESSIONID value. If omitted, the end-user cannot be charged and the client will not receive the kickback for these messages.

6.11 Originator (O)

The originator of the SMS message can be a numeric or alphanumeric string which will be displayed by the handset as the sender of the message. In case of a numeric originator, the maximum length is 16 digits. Alphanumeric originators have a maximum length of 11 characters.

Default value	Account default
Maximum length	11 characters or 16 digits

Parameter properties

6.12 Numeric originator (ONUM)

In case the originator is numeric, this parameter specifies whether the number is an international number or national shortcode. International numbers are displayed on the handset with a leading + character. If the client specifies a national shortcode as originator, this parameter should be given a value of 1.

Default value	0 (international number)
Value range	0 – 1

Parameter properties

6.13 Test request (TEST)

For testing purposes, this parameter can be used to simulate a request. In test mode, the gateway verifies all parameters but does not send any SMS messages. The gateway will return the same response in test mode as in normal mode.

Clients MUST NOT use this parameter to send large volumes of test requests.

Default value	0
Value range	0 – 1

Parameter properties

6.14 Scheduled delivery (DATE)

Messages can be scheduled for delivery at a future date/time. The parameter value depends on the value of the DATEFORMAT parameter (see 6.13). The gateway will queue the scheduled messages and delivers them at the specified timestamp.

Default value	Immediate delivery
Examples	cet: 2007-02-20 09:15:00 relative: 30 unix: 1162983334

Parameter properties

6.15 Scheduled delivery date format (DATEFORMAT)

The scheduled delivery timestamp (see 6.14) can be specified in different formats. The default value is **cet**, where the timestamp is specified as YYYY-MM-DD HH:MM:SS and indicates a timestamp in Central European Time format. The value **relative** means the timestamp is specified in seconds from the time the message is delivered to the gateway, whereas the value **unix** means the timestamp is specified in UNIX seconds since Jan 1st, 1970 at 00:00.

Default value	cet
Value range	cet / relative / unix

Parameter properties

6.16 Notification request (NOT)

For each sent SMS message the client can request a delivery notification. Delivery notifications are used to track the status of an SMS message. Each received notification will be automatically forwarded to the client via HTTP, SMTP, UCP, SMPP, CIMD2 or TCP. Notifications are enabled upon request only because of the necessary configuration details for the delivery to the client's system.

Default value	0
Value range	0 – 1

Parameter properties

6.17 Notification type (NOTTYPE)

When notifications are requested, this parameter defines which types of notifications are forwarded to the client. The following values can be specified:

Value	Type	Description
1	Accepted	Message accepted by destination SMSC
2	Buffered	Message buffered in destination SMSC for later delivery
4	Delivered	Message successfully delivered on the handset
8	Undelivered	Message could not be delivered to the handset

Table 6.2 - Description of the NOTTYPE values

A combination of values is also possible. If the client wants to request buffered and delivered notifications, a value of 6 should be specified. This is the sum of the buffered and delivered value.

Default value	12 (delivered + undelivered)
Value range	1 – 15

Parameter properties

6.18 Batch identifier (BATCHID)

Client can use a unique batch identifier to prevent duplicate deliveries of the same message. Certain network conditions can trigger a timeout when waiting for the gateway response while the message is already processed and sent by the gateway. If the client's application will retry the same request again, the message will be delivered multiple times to the specified recipient(s). When using a batch identifier, the gateway will check if the identifier has been used before. If so, the request will be rejected and no messages will be sent.

Default value	None
Maximum length	50 characters

Parameter properties

6.19 Message identifier (MSGID)

The message identifier is used for logging purposes and reporting. Clients can use message identifiers to separate different applications or customers in reports.

Default value	None
Maximum length	32 characters

Parameter properties

6.20 Validity (VAL)

The validity parameter, specified in seconds, determines how long an SMS message will be stored in the SMSC if the message cannot be delivered immediately (e.g. phone switched off). The SMSC will try to deliver the message using an operator defined retry scheme. After the specified validity period the SMSC will automatically delete the message from the queue and delivery to the handset will not take place.

Default value	172800 (2 days)
Value range	120 – 604800

Parameter properties

7. Gateway response

Each HTTP request from the client will be answered with an HTTP response. This response will contain standard HTTP headers and a response body. The content type of the response is ***text/plain***. The format of the response body is as follows:

```
[3-digit result code]=[additional information]
```

The possible result codes and their general meaning are as follows:

Result code	General meaning
0XX	Successful request, a total of XX SMS messages per recipient were sent
100 – 199	Rejected request
200	Internal error

Table 7.1 – Overview of result codes

7.1 Successful request (0XX)

If the request is successful, the gateway will return the amount of SMS messages sent per recipient. The additional information contains a unique identifier for this request. Clients can specify their own unique identifier by using the *BATCHID* parameter (see 6.18). If omitted, the gateway will generate a unique identifier of 16 digits. When the client requests delivery notifications (see 6.16), each generated notification will also contain the same identifier enabling clients to link the notification to the original request.

7.2 Rejected request (100 – 199)

A request can be rejected if parameters are either missing or invalid or if account settings do not permit the client to send messages. Rejected requests MUST NOT be retried by the client system, for as the same request will lead to the same result.

The additional information will contain a textual description of the error. For an overview of all rejected result codes and their description, please refer to Appendix A.

7.3 Internal error (200)

Internal errors are very rare and can occur in case of database connectivity problems. When a client receives an internal error, they SHOULD retry the same request after an interval of approximately 5 seconds to prevent flooding the server with requests.

8. Example request / response

Below is an example HTTP request for sending the message *Test Premium Message!* with an end-user of EUR 0.70 to two recipients *+31611111111* and *+31622222222*. The originator of the message is *My Service*. Please note the URL encoded characters in the N, M and O parameters.

```
POST /sendpsms HTTP/1.0\r\n
Host: gateway.wireless-services.nl\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 112\r\n
Connection: close\r\n
\r\n
VERSION=1.1&UID=username&PWD=password&N=31611111111%2C31622
222222&M=Test+Premium+Message%21&O=My+Service&RATE=70
```

The gateway will respond with the following HTTP response:

```
HTTP/1.1 200 OK\r\n
Date: Fri, 13 Oct 2006 16:05:38 GMT\r\n
Server: Apache\r\n
X-Gateway: Wireless Services\r\n
X-Gateway-URL: gateway.wireless-services.nl\r\n
Connection: close\r\n
Content-Type: text/plain\r\n
\r\n
001=1160938726272031
```

The result code is 001, indicating that the request was successful and that a total of 1 SMS message per recipient was sent. The unique identifier for this request is 1160938726272031.

9. Contact details

For more information about this specification, please contact our Support department.

Wireless Services B.V.
Newtonlaan 115
3584 BH Utrecht, The Netherlands

Telephone: +31 (0) 30 – 210 6499
Fax: +31 (0) 30 – 210 6666
E-mail: support@wireless-services.nl

10. References

HTTP	Hypertext Transfer Protocol – HTTP/1.1 http://www.w3.org/Protocols/rfc2616/rfc2616.html
MD5	Message-Digest Algorithm 5 http://www.ietf.org/rfc/rfc1321.txt
RFC2119	Key words for use in RFCs to Indicate Requirement Levels http://www.ietf.org/rfc/rfc2119.txt
SHA1	Secure Hash Algorithm 1 http://www.ietf.org/rfc/rfc3174.txt

11. Appendix A: Result codes

Table 11.1 lists all possible result codes which can be returned by the gateway, including the message types they are applicable to.

Result code	Message types	Description
0XX	SMS/MMS	OK, message accepted
100	SMS/MMS	Invalid interface version
101	SMS/MMS	Authentication failed
102	SMS/MMS	Invalid password format
103	SMS/MMS	Invalid authentication timestamp
104	SMS/MMS	Invalid authentication type
105	SMS/MMS	IP address not in whitelist
106	SMS/MMS	Blocked account, please contact support@wireless-services.nl
107	SMS	Notifications not allowed for this account
108	SMS	SMS Outbound not allowed for this account
109	SMS	SMS Outbound Premium not allowed for this account
110	MMS	MMS Outbound not allowed for this account
111	SMS/MMS	Duplicate batch identifier
112	SMS/MMS	No destination number(s) specified
113	SMS/MMS	Invalid destination numbers
114	SMS/MMS	Invalid group name

115	SMS/MMS	Insufficient SMS credits left
116	SMS	Alphanumeric originator too long
117	SMS	Numeric originator too long
118	SMS	Invalid validity
119	SMS	Validity out of range
120	SMS	Invalid data coding scheme
121	SMS	Invalid message class
122	SMS	Invalid protocol id
123	SMS/MMS	Invalid delivery timestamp
124	SMS	Invalid notification request
125	SMS	Invalid notification type
126	SMS	Message too long
127	SMS	Invalid binary user data header
128	SMS	Invalid binary message body
129	SMS	Empty message body
130	SMS	Invalid premium tariff value
131	SMS	Invalid premium operator code
132	SMS	Picture error
133	SMS	Ringtone error
134	MMS	Empty FROM parameter
135	MMS	Invalid SMIL URL
136	MMS	Error fetching URL

137	MMS	Missing required attribute
138	MMS	Maximum … parts in the message allowed
139	MMS	Maximum size of … KB exceeded
140	SMS/MMS	XML parse error
141	SMS/MMS	Invalid delivery timestamp format
200	SMS/MMS	Internal error, please try again

Table 11.1: Gateway result codes