

Interface description HTTP

Version 1.0

Document history

Version	Description
1.0	First release

1. Introduction

WhiteLabel Messaging Platform (MP) enables users to send message to and receive messages from mobile devices.

Outbound (mobile terminated) messages are sent from a Customer Application (Client) to MP using Hypertext Transfer protocol (HTTP).

Inbound (mobile originated) messages are sent from MP to Customer Server (Server) using HTTP.

Each message consists of a number of parameters sent as form parameters using HTTP POST or HTTP GET operations. POST is the recommended method since GET operations can be limited in size and may be truncated by browsers or intermediate proxies.

MP supports HTTP 1.0 and HTTP 1.1 with persistent connections.

1.1. SMS messages

SMS messages can be 1120 bits long. Depending on which character coding is used this limits the length of a message:

- GSM default 7-bit alphabet (IA5), max 160 characters
- 8-bit message, max 140 characters
- UCS2 (16-bit Unicode), max 70 characters

Sometimes messages include what is called UDH (User-Data Header). The UDH can contain information about special message properties and will limit the message length per SMS even further.

When a message submitted to MP is longer than the per message limit the message will be split into multiple SMS plus a UDH will be added indicating that each SMS is part of a longer message. The receiving handset will usually reassemble the message into the original “long” message.

1.2. Message encoding

Since SMS text messages usually are encoded using GSM 7-bit default character set (IA5) which is not very usual in other applications MP can make life easier and perform conversion between IA5 and ISO-8859-1. Whether conversion is performed or not is a configuration issue per user account.

When sending messages using UCS2 each SMS can hold a maximum of 70 characters. Longer messages will be split into multiple SMS.

1.3. URL encoding

All requests must be properly “URL encoded” as per RFC 1738. Since URLs can only include a limited set of characters (“A-Z”, “a-z”, “0-9”, “\$-_.+!*'(),”) all other characters needs to be properly escaped. There are utility classes that help the developer with this in most languages including Java, Perl, PHP and .NET.

Please note that source and destination addresses should never include a leading “+” character and also that a “+” in a URL is actually an “escaped” SPACE.

1.4. Addresses

Source and destination addresses (senders and recipients) are usually numeric international GSM numbers starting with country code, without leading zeros or plus signs. It is also possible to use alphanumeric source addresses which can be a maximum of 11 characters in the GSM network. Only use characters A-Z, a-z, 0-9 and space for best result.

When using threaded replies (SAT), the source address can also be an e-mail address and any replies will be delivered as e-mail to the specified source e-mail address. Please note that this requires the account is enabled for SAT and that replies is configured to be routed via e-mail/SMTP.

1.5. Delivery reports (DLRs)

When a message is sent to MP it is not known whether it can be delivered or not. However, it is possible to request a delivery report (using the parameter DLR=1) which will be submitted from MP to Server when the final status for the message has been reached. A DLR is structured as any other message but has a special message type set, it will also contain a reference to the id of the original message.

1.6. Threaded replies (SAT)

It is possible to send a message and receive a reply in which the original message id is referenced so that each reply can be uniquely mapped to the correct original message. MP makes this possible using Source Address Translation (SAT) so that the source address for each outbound message will be picked from a pool making the source+destination address combination unique for each message.

2. Client to Messaging Platform (sending)

This section describes how to send requests from the Client Application (Client) to the Messaging Platform (MP) and the possible responses to such requests.

The Client issues a HTTP GET or POST request to the MP using a specific URL. The MP issues back a response which completes the transaction. Please note that the MP response can either be an HTTP error code or a HTTP OK (200) in which case the returned response body will contain further information.

The Client supplies a number of parameters to the MP.

2.1. Submit message

2.1.1. Request parameters

Parameters are found in section 4, "Parameter reference".

2.1.2. Example sending text message

Username: user1
Password: secret
Destination address: +46-70-123456
Message: "Hello world"

```
http://sms.example.com:9011/bin/send?USERNAME=user1&PASSWORD=verysecret&DESTADDR=4670123456&MESSAGE=Hello+world
```

2.1.3. Example sending binary message

Username: user1
Password: secret
Destination address: +46-70-123456
Message: 0x41 0x42 0x43 0x4a

```
http://sms.example.com:9011/bin/send?USERNAME=user1&PASSWORD=verysecret&DESTADDR=4670123456&CHARCODE=2&MESSAGE=4142434A
```

2.1.4. Example sending binary message with UDH

Username: user1
Password: secret
Destination address: +46-70-123456
UDH: 0x06 0x05 0x04 0x0b 0x84 0x23 0xf0 (WAP push)
Message: 0x41 0x42 0x43 0x4a (not a valid message only to illustrate example)

```
http://sms.example.com:9011/bin/send?USERNAME=user1&PASSWORD=verysecret&DESTADDR=4670123456&CHARCODE=2&UDHI=1&MESSAGE=0605040B8423F04142434A
```

2.2. Responses

Responses from the Messaging Platform are HTTP responses which contain a status code which is either 200 (OK) or an error code. If HTTP status is OK then the response body includes further status information for the operation.

Response body include two or, optionally, three lines:

1. Contains the assigned message id, a numeric value, or -1 if the request failed.

2. Contains the status code, 0 for success or otherwise a numeric value other than 0 indicating what error occurred. Error codes: 1 – Unknown error, 2 – Syntax error, mandatory parameter missing, 10 – Access denied, 11 – Invalid message, 16 – No message credits left
3. If present, contains a text describing the error that occurred.

2.2.1. Example: Response indicating success

```
124365  
0  
OK
```

2.2.2. Example: Response indicating failure

```
-1  
2  
Recipient (DESTADDR) is missing
```

3. Messaging Platform to Client (receiving)

This section describes requests that are sent from the Messaging Platform (MP) to the Customer Server and how the Server should respond. Such requests are sent when incoming (Mobile-Originated) messages are received and forwarded to the customer. Same type of requests are used for Delivery Reports (DLRs).

MP issues a HTTP POST request to the Server using a specific URL. The Server must respond with a HTTP OK (200), body should be empty and will be ignored.

MP will supply a number of parameters. Parameters are found in section 4, “Parameter reference”.

Typical parameters for request containing message: ID, SOURCEADDR, SOURCEADDRTON, SOURCEADDRNPI, DESTADDR, MESSAGE, VP

Typical parameters for request containing DLR: ID, DLRID, SOURCEADDR, DESTADDR, STATUS, MSGTYPE, MESSAGE, VP

4. Parameter reference

Below the most common request parameters are listed. Other additional parameters may be present in requests. The rightmost column describes whether parameter is used for sending or receiving messages/DLRs.

Parameter	Description	Mandatory - M Optional - O	Send - S Receive - R
USERNAME	Username for MP account	M	S
PASSWORD	Password for MP account	M	S
ID	Message id	M	R
ORIGID	Original message id Only used for SAT when a reply to a previously sent message is received	M (SAT replies)	R
DESTADDR	Destination address (recipient) MSISDN should be on international format starting with country code. Example: 4670123456	M	S,R
SOURCEADDR	Source address (sender) Alphanumeric senders can be a maximum of 11 characters according to GSM spec only A-Z, a-z, 0-9 and space should be used	O	S,R
SOURCEADDRTON	Type of Number (TON) for source address Values: 0 Unknown 1 International (default) 5 Alphanumeric	O	S,R
SOURCEADDRNPI	Numbering Plan Indicator (NPI) for source address Should not be set when sending messages where it will default to 1.	O	S,R
DESTADDRTON	Type of Number (TON) for destination address Should not be set when sending messages where it will default to 1 (international number).	O	S,R

DESTADDRNPI	<p>Numbering Plan Indicator (NPI) for destination address</p> <p>Should not be set when sending messages where it will default to 1.</p>	O	S,R
CHARCODE	<p>Message encoding</p> <p>Values: 0 (GSM text, default also used for ISO-8859-15 when conversion enabled for account) 2 (Binary, 8-bit) 4 (UCS2, Unicode 16-bit)</p>	O	S,R
MESSAGE	<p>Message data</p> <p>Format depends on message type. For GSM text messages data is URL encoded. For binary or UCS2 messages data must be hex encoded.</p>	<p>M (messages) O (DLRs)</p>	S,R
MSGTYPE	<p>Message type</p> <p>1 Normal 5 DLR</p>	M (DLRs only)	R
DLR	<p>Request DLR</p> <p>Set to "1" to request that a delivery report (DLR) be sent back</p>	O	S
DLRID	<p>Original message id for DLRs</p> <p>When a DLR is received this parameter will contain the id of the original message it refers to</p>	M (DLRs only)	R
STATUS	<p>Status</p> <p>Indicates status of message</p> <p>1 – Delivered 2 – In process 3 – Failed 4 – Deleted 5 – Expired 6 – Rejected 7 – Canceled 8 – Queued 9 – Orphaned 10 – Relayed 11 – Unknown</p>	M (DLRs only)	R

KEYWORD	First word in message	O	R
VP	Validity Period Indicates how long (in seconds) a message will be stored for delivery before it will be expired	O	S,R
UDHI	User Data Header (UDH) Indicator When set to non-zero value, specifies that UDH is present in the message body.	O	S
SOURCEPORT	Source port Numeric value, 0 – 65535 This information will be added to UDH.	O	S
DESTPORT	Destination port Numeric value, 0 – 65535 This information will be added to UDH.	O	S
CONCATSMSREF	For concatenated messages Specifies a numeric reference number, 0 – 255 Not needed to send long messages.	O	S,R
CONCATSMSSEQ	For concatenated messages Specifies sequence number for this message part Not needed to send long messages.	O	S,R
CONCATSMSMAX	For concatenated messages Specifies total number of parts for this message Not needed to send long messages.	O	S,R