

# Premium SMS Platform Documentation

## Introduction to the Service

The Premium SMS platform provided by aql.com is a flexible way of receiving messages via short codes, identifying the source network provider and sending the appropriate billing messages to gain revenue.

## Connection Details

Our PSMS Platform is accessed via an Administration screen with requests being processed via our HTTP Post Interface:

```
http://gw1.aql.com/sms/gw-premium.php
```

or

```
https://gw1.aql.com/sms/gw-premium.php
```

Although we do offer the standard HTTP gateway, we do prefer that your application uses the HTTPS gateway as it prevents sensitive details being “sniffed” across the internet.

## Overview of Transactions

The basic process for making a billing request for a user is:

1. End user sends a text message to your short code
2. Our system records the message details and sends a HTTP Post request to your application with the message details and a unique identifier
3. Your application decides whether to bill the end user, if so, posting back to our gateway with a message to send back, as well as the unique identifier.
4. Once the message has been accepted, when we receive a delivery notification we can optionally post back to your application notifying you that the message was accepted / rejected from the phone

During step 3, when your application posts to our gateway it can flag whether to close use of the key again. This prevents further billing to this end user without an addition MO being received through the short code.

NB: Even if the message is accepted at the end users phone, it is still possible that payment will not be taken. This can be due to a variety of reasons, such as the end user rejecting the payment / the phone not having sufficient credit at the time the message is received.

In such a case, you will be charged 1 credit for an MT (Mobile Terminating) message.

## Gateway Details

When we receive a MO (Mobile Originated) message to a short code assigned to your account, we will send a HTTP Post to the location specified on the administration page on the aql.com site. This Post will contain the following details:

Variable Name	Description
request_type	This is the type of request, in this case it will be set to new
shortcode	The short code number
sender	The sender phone number
message	The message that was sent
timestamp	The timestamp of when the message was received
msgkey	A unique key associated with the MO message. This is needed for billing

Once your application has received these details, it can either ignore the request and not send back response thus not charging the end user, or send back a HTTP post back to our server requesting that the end user is charged (at the tariff associated with the shortcode). The details that your application would need to post back to our gateway are:

Variable Name	Description
username	Your aql.com username
password	Your aql.com password
shortcode	The shortcode associated with the request
destination	The destination number (Must be exactly the same as the number sent by us in previous request)
type	Must be set to either text or flash or wap_si
msgkey	msgkey sent with the original request
status	Final status of request, either open or close
report	Request delivery report, either yes or no
user_ref	This is an identifier for the request; it can be set to any alphanumeric string, up to 20 characters.

Depending on the type defined, the following fields may be required

For the types text or flash	
data	Message data (max 160 characters)

For the type wap_si	
wapname	The Name data used in the wap push
wapurl	The URL of the wap document
msg_limit	This is the number of messages the request can span (1, 2 or 3)

Note that if the status of the request is set to open, it allows further billing requests to be made using the same msgkey and destination number in the future. In this case, the user reference can be changed for each following charge request – this allows your application to identify each request (and associated delivery report).

However, if the status is set to close, this will prevent further billing attempts to be made using the msgkey and destination number combination. To allow further billing in this case, a new MO message must be received on the shortcode, giving a new msgkey.

When using the wap\_si type, the msg\_limit field is used to limit the number of billing messages sent, as if the combined length of the URL and name exceed 96 characters, the push will have to span 2 messages. If the message has to span 2 messages, 2 billing messages will be sent – hence charging the end user twice. The default limit is set at 1 message, and if the limit is reached, the error GW-MSGLIMIT\_REACHED will be given.

If report was set to yes, when we receive a delivery report back from the destination mobile, another request will be made to your application sending the following HTTP Post details:

Variable Name	Description
request_type	This is the type of request, this case it will be set to report
user_ref	This is the user_ref submitted in the original billing request sent by you
msgkey	A unique key associated with the MO message. This will allow you to match the report with the original request
status	The status of the message

## HTTP Gateway Error Codes

Below is a table of response codes from the http interface:

Error Code	Description
GW -NO_AUTH_DETAILS	No authentication details were provided
GW -AUTH_ERROR	There was an error in authenticating based on the details provided
GW -KEY_INVALID	The key provided was invalid (must only contain characters a-z,A-Z, 0-9)
GW -SHORTCODE_INVALID	The short code provided was formatted incorrectly (must only contain numbers 0-9)
GW -SHORTCODE_NOTFOUND	The short code was not found associated with your account
GW -SHORTCODE_INACTIVE	The Short code is inactive – contact support for details
GW -SHORTCODE_CLOSEDGW	The key for this transaction has been closed for use
GW -SHORTCODE_CLOSEDCLIENT	The key for this transaction has been closed by the client
GW -SHORTCODE_FAIL	Details for the short code could not be retrieved
GW-OK	Message Accepted
GW -FAIL	There was an error in queuing the message. Please contact support
GW -STATUS_INVALID	The status provided was invalid / missing
GW -USER_REF_INVALID	The User reference you provided was invalid / missing
GW -MSGLIMIT_INVALID	The value provided was not a valid number (1, 2 or 3)
GW -TYPE_INVALID	The message type was invalid (either text, flash or wap_si)
GW -DATA_INVALID	The data field was missing
GW -WAPNAME_INVALID	The wapname field was missing
GW-WAPURL_INVALID	The wapurl field was missing
GW -ENCODE_FAIL	The gateway failed to encode the binary message
GW -MSGLIMIT_REACHED	The binary message needed to span more messages than the limit set
GW-REPORT_INVALID	The report field was invalid (should be yes or no)

## Status Descriptions

Possible Values of the status variable are:

Status Name	Description
phonesent	The destination phone has accepted the message
phonefail	The destination phone has rejected the message
smscsent	The destination network has accepted the message on behalf of the destination number
smscfail	The destination network has rejected the message on behalf of the destination number
fail	The message has failed

## An example flow of requests during a billing transaction

This is the Message from the Mobile to the shortcode (this is a dump of the variables posted)

```
array(7) {
  ["request_type"]=>
  string(3) "new"
  ["shortcode"]=>
  string(4) "1000"
  ["sender"]=>
  string(12) "447789123456"
  ["message"]=>
  string(14) "This is a test"
  ["timestamp"]=>
  string(19) "2004-01-26 16:07:43"
  ["msgkey"]=>string(32) "966cf073923aa0793d1ecabbd10ba1bf"
}
```

Your application would then post back details similar to the following (NB:All fields are required), which would initiate the MT billing request

```
array(11) {
  ["username"]=>
  string(7) "testaccount"
  ["password"]=>
  string(5) "testpwd"
  ["shortcode"]=>
  string(4) "1000"
  ["destination"]=>
  string(12) "447789123456"
  ["type"]=>
  string(4) "text"
  ["data"]=>
  string(30) "This is the MT Billing message"
  ["msgkey"]=>
  string(32) "966cf073923aa0793d1ecabbd10ba1bf"
  ["status"]=>
  string(4) "open"
  ["report"]=>
  string(3) "yes"
  ["user_ref"]=>
  string(6) "123abc"
}
```

As in the MT billing request a status report was requested, when we receive any form of delivery confirmation, another HTTP Post would be made similar to the following:

```
array(4) {
  ["request_type"]=>
  string(6) "report"
  ["user_ref"]=>
  string(4) "123abc"
  ["msgkey"]=>
  string(32) "966cf073923aa0793d1ecabbd10ba1bf"
  ["status"]=>
  string(9) "phonesent"
}
```

At a later date, you may wish to bill the client again (e.g. a subscription service), you would need to send another MT Billing request, ie:

```
array(11) {
  ["username"]=>
  string(7) "testaccount"
  ["password"]=>
  string(5) "testpwd"
  ["shortcode"]=>
  string(4) "1000"
  ["destination"]=>
  string(12) "447789123456"
  ["type"]=>
  string(4) "text"
  ["data"]=>
  string(39) "This is the MT Billing message number 2"
  ["msgkey"]=>
  string(32) "966cf073923aa0793d1ecabbd10ba1bf"
  ["status"]=>
  string(5) "close"
  ["report"]=>
  string(3) "yes"
  ["user_ref"]=>
  string(6) "456abc"
}
```

In this case again, a status report was requested, therefore you would receive a similar delivery confirmation as before:

```
array(4) {  
  ["request_type"]=>  
  string(6) "report"  
  ["user_ref"]=>  
  string(4) "456abc"  
  ["msgkey"]=>  
  string(32) "966cf073923aa0793d1ecabbd10ba1bf"  
  ["status"]=>  
  string(9) "phonesent"  
}
```

After this, as the status was set to close, no further billing requests would be possible to this number using the msgkey. A new MO message would have to be received.

## Technical Support

For technical support and further queries about the Premium sms platform, please launch a support ticket, by going to the address:

[http://www.aql.com/account/aqs\\_index.php](http://www.aql.com/account/aqs_index.php)

using your aql.com username and password to login.