



## Send SMS legacy API

Our legacy API still allows users to send their messages via HTTP/HTTPS.

**This SMS API is deprecated. This has been superseded by the aqi RESTful API. Documentation can be found at <https://api.aqi.com/doc/>**

Our legacy API still allows users to send their messages via HTTP/HTTPS.

```
curl -d to_num=447778123456 -d message=hello+world  
http://username:password@gw.aqi.com/sms/postmsg.php
```

Here is an [example PHP script](#) an [example ASP script](#) and an [example Shell script](#) for connecting to our gateway\* (Shell script kindly written by Ethan Galstadt of [Nagios](#) for aqi.com. Thanks Ethan!). There is also a [PEAR module](#) under development. This includes basic functions such as message sending and also advanced functions such as address book definitions. Example of simple [concatenated SMS form](#) which uses our HTTP POST API. \*note - the [bash script](#) is also a notification script which can be used by Nagios users.

For more information, please see [here](#).

An example using [CURL](#) is given below, which can be incorporated as the posting agent within a PHP or Perl script. On the BASH/CSH/KSH/ZSH command line (Linux/SunOS) with CURL installed type.

Or, using [LYNX](#) which is also standard with most installs of [RedHat](#).

```
lynx -dump -auth=username:password  
"http://gw.aqi.com/sms/postmsg.php?to_num=447778123456&message=hello+world"
```

If you'd like us to put together a quotation/specification for an SMS or database driven project, please [contact us](#).

The syntax for the request is as follows (note: this gateway supports both POST and GET:

### Via standard port 80

```
http://gw.aqi.com/sms/postmsg.php?to_num=447778123456&message=test+message+ok  
&flash=1&username=username&password=password
```

or

```
http://username:password@gw.aqi.com/sms/postmsg.php?to_num=447778123456&message=test+message+ok&flash=1
```



## Via SSL port 443

`https://gw.aql.com/sms/postmsg.php?to_num=447778123456&message=test+message+ok  
&`

`flash=1&username=username&password=password`

or

`https://username:password@gw.aql.com/sms/postmsg.php?to_num=447778123456&mes  
sage=test+message+ok&flash=1`

## Definitions

username = username of your [aql account](#).

password = password of your [aql account](#).

to\_num = mobile number of recipient - entered in international format (e.g. 07778123456 would be entered as 447778123456 - omit the leading zero and add the country code for the UK which is 44).

message = message string (separate each word by a "+" sign to indicate a space).

flash = this field is optional, set flash=1 to send the message as a flash text message. This means that the message will be displayed on the recipient's phone without the phone announcing the message or storing it.

orig = this field is optional. Set orig=newname to set your originator to a specific originator name for that message. Omitting this field will mean that your default originator is used.

sendtime = this field is optional. set sendtime="time to send" - this will put the message in our queues scheduled to send on the time and date specified. The syntax is MySQL date format - e.g. 2006-06-01+13:59:00 for 1.59pm on the first of June 2006.

## Server response codes

When an SMS request is posted to our gateway, it will issue one of the following responses, which you can log or echo to your client or trigger an event.

### *Response*

Status	Description
--------	-------------

AQSMS-NOAUTHDETAILS	The username and password were not supplied
AQSMS-AUTHERROR	The username and password supplied were incorrect
AQSMS-NOCREDIT	The account specified did not have sufficient credit
AQSMS-OK	The message was queued on our system successfully
AQSMS-NOMSG	No message or no destination number were supplied
AQSMS-CREDIT	The output if the string cmd=credit is added to the query - allowing users to track their account balance via our http interface: e.g. below

<http://gw.aql.com/sms/postmsg.php?username=yourusername&password=yourpassword&cmd=credit>

This means that you can use the aql textback service and log the clients' data via your own server and database rather than the data being stored on our system. We can also offer support and consultancy on this subject. In addition, we can accept your SIM cards for incorporation into our modem banks. This will allow us to accept text messages on your own private number without the need for a keyword. Contact us for prices.

### Multiple message gateway

Our multiple message gateway allows a single message to be sent to multiple mobiles in a single HTTP request. The address of the gateway is:

<http://gw.aql.com/sms/postmsg-multiple.php>

The syntax for the request is identical to the above except in the following regards:

#### *Definition modification:*

to\_num = comma separated list of mobile numbers.

Please make sure the whole list contains no spaces e.g.

to\_num=4477111111,4477222222,4477333333

#### *Server response code modifications:*

AQSMS-OK:x = From the list above, x messages were queued on our system successfully.

This response code will be received in place of AQSMS-OK

If any of the numbers are invalid, this gateway will still send messages to the valid numbers.

The 'x' reflects the number of VALID mobile numbers that messages were queued for. It is recommended that all mobile numbers be validated before they are sent to the gateway.

## Delivery notification via HTTP gateway

When sending messages via our systems, it is possible to track the delivery status of each message. This can be accomplished by appending an extra variable to the postmsg.php request: Along with setting the username, password, destination number, message (and optionally the originator), the variable "dlrurl" can be specified which includes an URL Encoded address of a script on your server, with 2 optional parameters included. An example of this use is shown below:

Sending a message with the following details:

```
username=test
password=test
to_num=447740123456
orig=aql
message=this is my message
dlrurl =
http://test.aql.com/deliveryreport.php?reportcode=%code&destinationnumber=%
dest&myreference=123
```

The dlrurl has two special values in it - %code and %dest - these are substituted for the real report code and destination number. Any other variables can be set by you for your own tracking of the message submitted, for example myreference in this case.

%dest = destination number of message

%code = delivery report code

The %code values are:

- 1 = Delivered to handset
- 2 = Rejected from handset
- 4 = Buffered in transit (phone probably off / out of reception)
- 8 = Accepted by SMSC
- 16 = Rejected by SMSC

All of the details of the message have to be encoded for sending via a HTTP GET request, so put together - the final request looks like:

```
http://gw.aql.com/sms/postmsg.php?
username=test&password=test&to_num=447740123456&orig=aql&
message=this+is+my+message&dlrurl=http%3A%2F%2Ftest.aql.com%2Fdeliveryreport.php
%3Freportcode%3D%25code%26destinationnumber%3D%25dest%26myreference%3D123
```

Once the message gets processed, our system would simply call the address set by dlrurl, substituting %code for the delivery code value and the same for the %dest value. So, if the



message were successfully delivered to the handset, your system would receive the following request:

<http://test.aql.com/deliveryreport.php?reportcode=1& destinationnumber=447740123456>