QUIOS®

GLOBAL MOBILE MESSAGING

Q200 API Documentation

SMS GATEWAY SERVICES

Friday, January 30, 2004

custsupport@quios.net

Table of Contents

The client should HTTP POST request at
q200gateway.quios.com/services/smsgateway

All HTTP requests are required to have the following authentication parameters:

Common Fields	Valid Values
username	in email format e.g. ronald@quios.com
password	The password in
version	1.0 is the current version

1. Sending SMS

Aside from authentication parameters, additional parameters are required to send an SMS. They are:

Additional parameters:	Valid Values
test_mode	yes or no
type	The message type. It supports GSM0338, UCS2, RTTTL and Binary
text	the body of the message
numbers	destination numbers separated by comma (,).
originator	sender's originator either alphanumeric or numeric. Numeric originator should be activated at http://q200.quios.com

1.1 Response:

The generated content type of the HTTP response is in text/plain body. The possible responses are as:

```
<response> ::= <response_code> { <CR_LF> <response_code> }
<response_code> ::= <destination_number> <delimiter> <success>
| <failure>;

<success> ::= '0' <delimiter> <balance> <delimiter> <charge>;
<balance> ::= <integer>;
<charge> ::= <integer>;

<failure> ::= '1'<delimiter><failed_code> {<cr_lf> <failure>}
<failed_code> ::= <general_failure> | <user_password_mismatch>
|
        <account_check_failure> |
        <message_limit_exceeded> |
        <wrong_number_length> | <invalid_originator>|
        <mssing_attribute> | <unsupported_version> ;

<general_failure> ::= '1 ' <delimiter> <string>;
<missing_attribute> ::= '2' <delimiter> <quote>
                        'USERNAME' | 'PASSWORD' |  'TEXT' |
                        'ORIGINATOR' | 'NUMBERS' | 'METHOD' |
                        'VERSION'<quote>
<user_password_mismatch> ::= '4';
<message_limit_exceeded> ::= '8' ;
<account_check_failure> ::=
'16'<delimiter><balance><delimiter>
                                <charge> ;
<unsupported_test_mode> ::= '32' ;
<unsupported_version> ::= '64' <delimiter> <string>
<wrong_number_length> ::= '128' <delimiter><string>;
<invalid_originator> ::= '256' <delimiter><string>;
<delimiter> <string> ;
<integer> ::= /* positive integer */ ;
<string> ::=  /* character string */ ;
<delimiter> ::= ','
<quote> ::= '"';
<cr_lf> ::= CR LF;
```

1.2 Examples:

1.2.1 Successful request on sending

For successful request on sending, the response would be like:

`639164408740,0,6940,40`

Where `639164408740` is the destination number; `0` is the code for successful request; `6940` is the remaining credit; `40` is the charge or the credits used.

1.2.2 Successful sending to multiple recipients

For successful sending message to multiple recipients, the response would be the same format as single recipient with the additional response separated by new line or <CR><LF>

`639164408740,0,6940,40<cr><lf>`

`639161234567,0,6900,40<cr><lf>`

1.2.3 Failed request

For failed requests, the response would be like:

```
639164408740,1,4
```

Where `1` means the code for failed status and `4` is username and password mismatch.

1.2.4 General failure

For failed request with general failure, the response would be like:

```
639164408740,1,1,"Service Unavailable"
```

or

```
63916,1,1,"err_number_wrong_length"
```

Where `1` in the 3rd column means the failure code of general failure, and the 4th column is the reason of the failure of the request.

1.2.5 Failed request with multiple recipients

For failed request with multiple recipients, the response would be like:

```
63916,1,1,"err_number_wrong_length"
639164408740,1,1,"Service Unavailable"
```

2. Querying Registered Originators

To get the mobile numbers activated, only authentication parameters are required.

The response definition is:

```
<originator_synchronize_request> ::= <originator_number> {<
originator_number > <cr_lf >} | <failed_code>;
<failed_code> ::= <username><delimiter><handle_do_not_exist> |
                  <handle_password_mismatch> |
                  <originator_do_not_match> |
<missing_attribute> |
                  <service_not_available> |
<unsupported_version> |
<handle_do_not_exist> ::= '1';
<handle_password_mismatch> ::= '2';
<originator_do_not_match> ::= '4';
<unsupported_version> ::= '8';
<missing_attribute> ::= '16';
<service_not_available> ::= '32'
<originator_number> ::= <numeric> /* the originator number */
<cr_lf> ::= CR LF
```

1.3 Examples:

1.3.1 Success

For successful querying originator, the response just returns the list of mobile numbers activated separated by line or CR LF:

```
639164408740
```

```
63916712345
```

1.3.2 Authentication error

For authentication error:

```
2
```

…

3. Querying remaining Credits

To get the remaining credits, only authentication parameters are required.

The response definition is:

```
<response> ::= <response_code> { <CR_LF> <response_code> }
<response_code> ::= <success> | <failure>
<success> ::= <username><delimiter>
              <response_code><delimiter><balance>;
<failure> ::= <username_not_found> | < missing_attribute > |
              <service_not_available> ;
<username_do_not_exist> ::= '1';
<username_password_mismatch> ::= '2';
<missing_attribute> ::= '4'
<service_not_available> ::= '8';
```

1.4 Examples:

1.4.1 Successful query

For successful query for balance:

```
ronald@quios.com,0,70335
```

Where `ronald@quios.com` is the username; `0` is the code for successful request and `70335` is the number of credits remaining.

1.4.2 Authentication error

For authentication error:

```
marc@quios.com,1,2
```