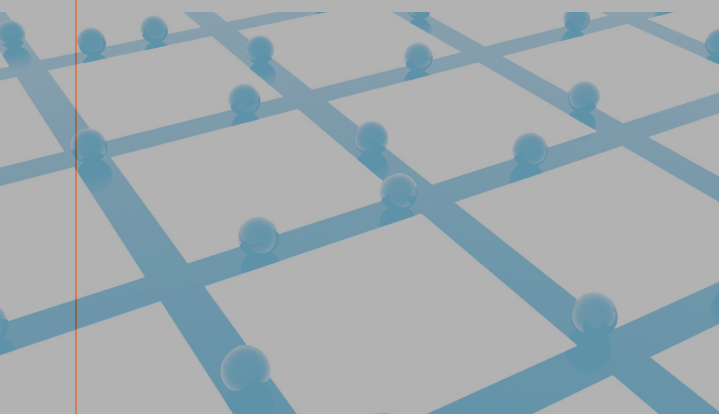


HLR LOOKUP

User guide



The information contained in this document is proprietary and copyright and for the sole purpose of informing customers about the above service. The service is owned by Routo Telecommunications Ltd, 2nd Floor, Kingsgate House, 115 High Holborn, London, WC1V 6JJ, United Kingdom.

www.routomessaging.com
support@routotelecom.com



Associate
Member

**Your Success.
Delivered.**

HLR LOOKUP

User guide

Preface:

Please register for [free SMS and HLR Lookup account](#) for testing our SMS, MMS and HLR services and integration to our SMS, MMS, HLR APIs.

Within the same account you can use the HLR Lookup alongside SMS delivery or as a separate service. The following features and services are available:

- Straightforward, SMS API, MMS API and HLR API integration:
 - Send and receive SMS and MMS using HTTP and SMPP
 - Perform HLR Look Up's either individually or in bulk via HTTP API or Web browser
- Free sample code on how to send, receive SMS and MMS and perform HLR lookups
- Status and the real time delivery reports of the messages you have sent
- Rapid and accurate HLR lookup results returned through the HTTP callback request
- Free [Support](#), manned and around the clock 24/7 to answer any questions
- Minimal SMS, MMS and HLR development time
- Deploy with confidence; we have over ten years of experience in Mobile Messaging Integration

Table of contents:

1	CHANGE HISTORY	3
2	INTRODUCTION	4
3	COVERAGE.....	6
4	HOW TO USE THE SERVICE.....	7
5	API DATA EXCHANGE FORMATS	8
6	HOW TO MAKE ONE AT A TIME REQUESTS.....	9
6.1	Response data	10
6.2	HTTP/S GET request example	11
6.3	Perl code sample for sending single requests	12
6.4	PHP code sample for sending single requests	13
6.5	Error messages for the request.....	14
7	HOW TO MAKE A BULK REQUEST	15
8	RECEIVING HTTP CALLBACK	16
8.1	Perl code sample for callback processing	17
8.2	PHP code sample for callback processing	17
9	FEEDBACK.....	18

List of code snippets:

Snippet 1: PERL code sample for sending the requests one at a time	12
Snippet 2: PHP code sample for sending the requests one at a time	13
Snippet 3: Perl code sample for callback processing	17
Snippet 4: PHP code sample for callback processing	17

List of tables:

Table 1: Change history	3
Table 2: Lookup parameters	9
Table 3: Status field values	9
Table 4: Response data	10
Table 5: Bulk lookup parameters	15

1 Change History

Version	Date	Author	Section	Changes to documentation
1.1	29-June-2009	Aleksandar Petrovic		

Table 1: Change history

2 Introduction

The purpose of this document is to describe the Routo Messaging HLR Lookup service. The intended audiences of this document are the developers and end users who will use our HLR Lookup service to search for the mobile phone numbers and additional information about the owners of those numbers i.e. mobile network users.

The HLR (Home Location Register) is the main database of subscriber information for a mobile network. Depending on the network operator, the HRL can contain the following information about the users:

- user information
- account status
- location

The Routo Messaging HLR lookup technology allows you to look up the phone numbers within multiple Home Location Registers (HLR).

The search can be performed by using one of the three following methods:

- **by using the HTTP API** – the lookup is performed in the backend by using the HTTP API (see section 0)
- **by using your browser** – you can use the [RoutoMessaging Command Centre](#) to upload the list of numbers you want to lookup which will be filtered/cleansed and then you will receive the results back
 - **note:** this feature is very useful for Marketers who want to check if the numbers are valid prior to sending the messages to the users
- **by CSV file upload** – from our Command Centre

The responses contain MCC and MNC codes, which identify the country and the network the queried number currently resides on. In some cases, the response may contain additional fields if those are supported by the operator/network.

These are the possible types of information returned as a result of a query:

- **MCC / MNC Codes** - by having the MCC / MNC information, you can determine whether the subscriber has ported his / her number to a different network compared with the operator prefix (also known as line range) of the number. A Mobile Network Code (MNC) is used in combination with a Mobile Country Code (MCC) (also known as a .MCC / MNC tuple) to uniquely identify a mobile phone operator/carrier using the GSM, CDMA, iDEN, TETRA and UMTS public land mobile networks and some satellite mobile networks. The ITU-T Recommendation E.212 defines mobile country codes.
- **IMSI** - our HLR service can return the 15 digit IMSI (International Mobile Subscriber Identity number) of the mobile subscriber's phone number (MSISDN). The IMSI is a unique number which identifies the mobile subscribers in GSM and UMTS networks. This IMSI information is retrieved directly from the SS7 network. Querying the IMSI provides subscriber information for authentication and authorization of the user in mobile security applications and scenarios, such as in online banking. The knowledge on IMSI number allows also enables certain Push to Talk (PTT) services.
- **MSC** - MSC is short for Mobile Switching Centre; retrieving the MSC information enables you to determine the approximate location of the mobile user.

- **Status** - the following are the examples of possible statuses:
 - OK
 - ERROR
 - UNKNOWN SUBSCRIBER
 - ABSENT SUBSCRIBER
 - FACILITY NOT SUPPORTED
 - TELE SERVICE NOT PROVISIONED
 - NETWORK ERROR

There are three modes for requesting information:

- **one at a time (synchronous) mode** – when using this mode the response is received immediately
- **bulk (asynchronous) mode** – if the lookup is performed by using the bulk mode, you can send up to 100 numbers per message, and receive response through a HTTP callback after the server has finished processing your requests
- **CSV upload** – from our Command Centre

RoutoMessaging HLR Lookup helps different types of customer achieve **great cost savings**:

- Marketers - clean up those lists or database of mobile numbers - filter out invalid numbers.
- Operators, aggregators and service providers - avoid paying unexpected interworking fees for ported numbers, route voice and sms traffic through the least cost provider (save costs on correct routing and billing)

Main features:

- Immediate service activation
- Fast query response
- HLR interface - HTTP API or Web browser
- Pay per query based charging
- Use the same account and balance as for your other RoutoMessaging services
- Coverage of all mobile numbers in the worldwide countries

3 Coverage

For detailed information about all operators and numbers the service covers, please visit the HLR coverage page within [Command Centre](#).

Note: The coverage will progressively expand. For any other destinations not listed, please contact your Account Manager.

We are available at following:

Email:

- For sales enquiries from South East Asia sales.asia@routotelecom.com
- For sales enquiries from all other regions: sales@routotelecom.com
- Support: support@routotelecom.com
- Billing: accounts@routotelecom.com
- Customer Care: customercare@routotelecom.com
- Careers: careers@routotelecom.com
- Trading: trading@routotelecom.com

Telephone

- Switchboard: +44 (0) 870 231 7777
- Sales and General Enquiries +44 (0) 870 231 7777
- Support (manned 24x7) +44 (0) 870 231 7777
- Fax +44 (0) 870 231 7775

United Kingdom (headquarters)

Routo Messaging
Routo Telecommunications Ltd
2nd Floor, Kingsgate House
115 High Holborn
London, WC1V 6JJ
United Kingdom

4 How to use the service

RoutoMessaging provides a simple API which is used to query the Home Location Register (HLR). The responses from the HRL contain MCC/MNC (MCC and MNC codes), which identify the country and the network the queried number currently resides on. Depending on the operator/network, other fields may also be available.

There are three modes of requesting information:

- **one at a time (synchronous) mode** – this mode allows you to receive the response right away
- **bulk (asynchronous) mode** – this mode allows you to send up to 100 numbers per message and receive the response through an HTTP callback after the server has finished processing of your requests.
- **using our Command Centre** where you will be able to do one at a time HLR lookup using our web interface, or bulk by uploading the list of numbers in csv format

5 API Data exchange formats

The way you will send your data is by using the POST/GET HTTP requests, and our responses will be in JSON format which allows simple parsing.

JSON is a lightweight computer data interchange format. There are parsers/generators (bindings) for most programming languages, so you will not need to write your own parsing code to process our responses. More information about JSON is available at:

- <http://json.org/>

Important note: we strongly encourage you to use the existing JSON parser instead of writing a custom one.

6 How to make one at a time requests

In order to make a one at a time request, an HTTP request (GET or POST) should be sent to <http://hlr.routotelecom.com/> and it must contain your user name, password and one number to look up.

The following are the parameters for the number lookup:

Value	Description
user	your RoutoMessaging account user name
pass	your RoutoMessaging account password
number	It should contain only one number. Number should be in full international format WITHOUT + or 00 at front.

Table 2: Lookup parameters

The server response will be in JSON format. Depending on the queried operator/network, it is possible that not all of the data fields will be populated.

However, the minimum information received from the server in case of the OK response will be number, MCC and MNC fields. Other fields will be populated where available.

If the server accepts your request, it will respond with the status OK and response data. Below is the sample of the server response.

```
{"status":"OK","number":"123456789","imsi":"220021234567890","mcc":"220","mnc":"02","msc":"13245100001","msc_location":"London,UK","operator_name":"Orange (UK)","operator_country":"UK"}
```

In case the server rejects your HLR lookup, the status will be set to ERROR and you will receive the error description inside the message field. Below is the sample of the server response.

```
{"status":"ERROR","message":"INVALID REQUEST - NO NUMBER GIVEN"}
```

Other values besides the OK and the ERROR status values are possible as well. For instance, a different status value can be returned upon a successful query, in case the queried number was invalid. All possible values for the status field are listed in the table below.

Value	Description
OK	If HLR lookup is successful – you can expect at minimum number, mcc and mnc fields to be populated.
ERROR	If there was an ERROR in processing the request – error details will be within message field.
UNKNOWN SUBSCRIBER	The number is invalid or not assigned to anyone.
ABSENT SUBSCRIBER	The cell phone is turned off (not available).
FACILITY NOT SUPPORTED	The number does not support SMS.
TELE SERVICE NOT PROVISIONED	Sim card hasn't been activated yet
CALL BARRED	The SMS service is barred by the operator
NETWORK ERROR	Lack of the response from network

Table 3: Status field values

We are working on adding additional fields for the response and you will be notified of any such change. Please note that since we are using the JSON data exchange format you will only need to modify your code if you want to utilize new fields.

Your existing code will continue to work without the need for modification but will not retrieve the information about the newly added fields.

6.1 Response data

Currently, the following fields will be returned with the OK response. Depending on the queried network, some fields might not be populated, but the response will contain the following information as a minimum:

- Status
- Number
- MCC and MNC

Unpopulated fields will be set to null value per JSON standard.

Value	Description
status	Will be set to OK
number	The number you are doing HLR lookup for
imsi	International Mobile Subscriber Identity
mcc	Mobile Country Code
mnc	Mobile Network Code
operator_name	Name of operator
operator_country	Country of operator
msc	MSC Location
msc_operator	The name of the operator where the number is located (roaming)
msc_mcc	Mobile Country Code of operator where number is located
msc_mnc	Mobile Network Code of operator where number is located
msc_country	The country of the operator the number is located in
msc_location	The location of MSC (city or region – depending on operator)

Table 4: Response data

We are working on adding additional fields for the response and you will be notified of any such change. Please note that since we are using the JSON data exchange format you will only need to modify your code if you want to utilize new fields.

Your existing code will continue to work without the need for modification but will not retrieve the information about the newly added fields.

6.2 HTTP/S GET request example

This section of the document will give you an example of the HTTP GET request.

<http://hlr.routotelecom.com/?user=username&pass=password&number=123456789>

If you try entering the URL above in your web browser, it might present you with a save as (open with) window. This is normal because we are responding with an application/json header and the response data in JSON format.

You can save the file and open it with any text editor to view the response. In reality you would be either using our web interface, or calling that URL from your application.

For a higher level of security, our customers can use an SSL connection for querying an HLR either for single or bulk lookups. In this case, sample HLR request should look like this:

<https://hlr.routotelecom.com/?user=USERNAME&pass=PASSWORD&number=123456789>

Perl code sample for sending single requests

The snippet below demonstrates the Perl code sample for sending the single requests i.e. sending the requests one at a time rather than as a bulk request.

```
#!/usr/bin/perl
use strict;
use CGI;
use JSON;
use Data::Dumper;
use LWP::UserAgent;

my $q = CGI->new();
print $q->header();

my $ua = LWP::UserAgent->new();
$ua->agent('RoutoStarAsk/1.0');

my $dest = '44123456789';
my $user = 'username';
my $pass = 'password';

my $res = $ua->post(
"http://hlr.routotelecom.com/",
{
    number => $dest, user => $user, pass => $pass
}
);
if ($res->is_error()) {
    # couldn't make request
    print "Couldn't make request: $res->status_line";
}
else {
    my $resp = from_json($res->content());
    print '<pre>' . Dumper($resp) . '</pre>';
}
```

Snippet 1: PERL code sample for sending the requests one at a time

CGI, Data::Dumper and LWP::UserAgent are Perl modules that are a part of the standard Perl distribution, while JSON needs to be installed.

6.3 PHP code sample for sending single requests

The snippet below demonstrates the PHP code sample for sending the single requests i.e. sending the requests one at a time rather than as a bulk request.

```
<pre>
<?php

$dest = '44123456789';
$user = 'username';
$pass = 'password';

$ch = curl_init("http://hlr.routotelecom.com/");
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,
"number=$dest&user=$user&pass=$pass");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$output = curl_exec($ch);
curl_close($ch);

echo print_r(json_decode($output), 1);

?>
</pre>
```

Snippet 2: PHP code sample for sending the requests one at a time

The PHP needs to be compiled with the support for CURL and JSON. On computers using Windows operating system, it is usually enough to just enable those extensions within the php.ini file. Alternatively you may use some of PHP's JSON libraries linked from <http://www.json.org>.

This sample code only sends the request and displays the results. When using your application you will probably want to process the results and store them in DB.

6.4 Error messages for the request

In case the received response is ERROR, the following error messages are generated and are contained within the message field.

- **LOGIN FAILED** - the user name and password you are sending with your request are not matching the ones in our database, please double check that you entered the correct info before contacting our support
- **INSUFFICIENT CREDIT TO PROCESS YOUR REQUEST OR YOUR ACCOUNT MAY BE DISABLED** - your account is disabled – possibly due to low balance, please contact our support team
- **LOGIN INACTIVE** - your account is disabled – possibly due to low balance, please contact our support team
- **SERVICE NOT ENABLED FOR YOUR ACCOUNT** - number lookup service has not been enabled for your account, please contact your account manager at Routo
- **NO NUMBER SENT FOR LOOKUP (one at a time request only)** - your request does not contain a number. Please check your request.
- **NO NUMBERS SENT FOR LOOKUP (bulk request only)** - your BULK request does not contain any numbers. Please check your request.
- **TOO MANY NUMBERS SENT FOR LOOKUP (bulk request only)** - you sent more than 100 numbers in your BULK request – please send only up to 100 numbers per request.
- **SERVICE NOT CONFIGURED FOR YOUR ACCOUNT (bulk request)** - the HLR Look Up service has not been fully configured for you, please contact our support and have at hand the URL that our callback application will call on your server to give you back the results.
- **LOW BALANCE FOR REQUEST** - Your account balance does not have sufficient funds for the request to be accepted. You should make a payment towards your account balance. If you are a prepay customer, or contact RoutoMessaging support if you are a post pay customer.

7 How to make a bulk request

This section of the document will give you instructions for sending a bulk request. Beside the one at a time request previously described, you can also send a bulk HTTP request.

This method allows you to submit up to 100 numbers for lookup, using one HTTP request. You will only receive the status and the message fields (in JSON format) for this request. In addition either none or all numbers in bulk request are accepted for processing.

In order to create a bulk request, you need to send an HTTP request (GET or POST) to this URL: <http://hlr-bulk.routotelecom.com/> which must contain the following:

- Your username
- Your password
- Between 1 and 100 numbers to look up

The following are the parameters for the BULK lookup:

Value	Description
user	your RoutoMessaging account user name
pass	your RoutoMessaging account password
numbers	It should contain at least one number, and not more than 100 numbers separated by comma (,) sign. Each number should be in full international format WITHOUT + or 00 at front.

Table 5: Bulk lookup parameters

If the server accepts your request, it will respond with the status OK. Please note that the HLR response for each of the requested numbers will be provided through a callback, so no additional data is provided here in contrast to one by one request.

```
{"status":"OK","message":"REQUEST ACCEPTED"}
```

In case the server does not accept your request, you will receive the ERROR status and the error details within the message field.

```
{"status":"ERROR","message":"TOO MANY NUMBERS SENT FOR LOOKUP"}
```

8 Receiving HTTP Callback

The Bulk HLR Look Up returns the results to you through a separate HTTP request, a feature termed as “callback”. The callback will make a POST HTTP request back to your server, containing the result data in JSON format within the *response* form field.

The result data will contain the result of at least one HLR lookup, and at the most it will contain 50 results. If there are more than 50 results, you will receive several callbacks.

The callback sends all of the results that are available at the time the particular callback is made – if not all of your requests have been processed yet, you may receive results through several callbacks.

The format/fields are the same as for one by one request – and you will receive an array of 1 to 100 such results. Sample containing two HLR results:

```
[{"status": "OK", "number": "123456789", "imsi": "220021234567890", "mcc": "220", "mnc": "02", "msc": "13245100001", "msc_operator": "Orange (UK)", "msc_mcc": "220", "msc_mnc": "02", "msc_country": "United Kingdom", "msc_location": "London", "operator_name": "Orange (UK)", "operator_country": "UK"}, {"status": "OK", "number": "123456789", "imsi": "220021234567890", "mcc": "220", "mnc": "02", "msc": "13245100001", "msc_operator": "Orange (UK)", "msc_mcc": "220", "msc_mnc": "02", "msc_country": "United Kingdom", "msc_location": "London", "operator_name": "Orange (UK)", "operator_country": "UK"}]
```

The callback only checks the HTTP status (200 means OK) without any regards to the text content of your response.

If the callback receives any other status but 200/OK, it will pause for 5 seconds before trying to deliver the response to you again. It will try 3 times to deliver the response to your server. After 3 unsuccessful attempts the request will be marked as undeliverable and the callback will not try to deliver those HLR results again.

8.1 Perl code sample for callback processing

The following snippet contains a Perl code sample for the callback processing.

```
#!/usr/bin/perl
use strict;
use CGI;
use JSON;

my $q = CGI->new();
my $form = $q->Vars();

my $response = $form->{response};

if(defined($response) && length($response)){
    use Data::Dumper;
    open(LOG,">>./receive_callback.log") || die "Could not open log file:
    $!";
    print LOG Dumper( from_json($response) ), "\n\n";
}

print $q->header(), "OK";
```

Snippet 3: Perl code sample for callback processing

Note: this example only writes the results in a text log file. However, you may process the received data as necessary.

8.2 PHP code sample for callback processing

The following snippet contains a PHP code sample for callback processing.

```
<?php
$response = $_REQUEST['response'];
if($response){
    $fh = fopen('./receive_callback.log', 'a') or die("Can't open
    file");
    $resp = json_decode($response);
    fwrite($fh, print_r($resp, 1) . "\n\n");
}
echo "OK";
?>
```

Snippet 4: PHP code sample for callback processing

Note: this example only writes the results in a text log file. However, you may process the received data as necessary.

9 Feedback

We value your feedback on this document. If you believe any detail is missing, please contact our Support team.

- **Email** support@routotelecom.com
- **Telephone** +44 (0) 870 231 7777
- **Live Chat** is accessible from:
 - “**Help Section**” of the [Command Centre](#)
 - or within the “**Live Support**” section at www.routomessaging.com