

viahub HTTP API documentation

Table of Contents

Introduction.....	2
Sending GET (query string) Request.....	3
Sending POST (XML) Request.....	4
Responses.....	5
Message Parameters.....	6
Message Types.....	8
Receiving delivery reports.....	9
SMS Routes.....	10
Mobile Number Portability (MNP) Check.....	11

Introduction

In order to use the viahub HTTP API, simply send a GET or POST request to our Gateway interface.

When sending GET request, you can make a query string request.

When sending POST request, you can:

- provide a JSON object
- provide an XML document

All requests must be sent to the following URLs:

Request type	URL
GET (query string)	https://gateway.viahub.net/single
POST (JSON)	https://gateway.viahub.net/json
POST (XML)	https://gateway.viahub.net/xml

Please note: when sending a message via a GET request, only **one** message can be sent per HTTP request, whereas you can send one or more messages with a POST request, **up to the limit of 100 messages per request**.

Sending GET (query string) Request

Sending a single message via GET request to our API couldn't be simpler. All you have to do is specify the message parameters and you're good to go. Please check "Message Parameters" section for more info on what parameters are available.

Example

```
/single?  
api_key=my_very_very_secret_api_key12345&from=14155992670&to=141559  
92671&body=Hello%20World!
```

This would send a "Hello World!" message from 14155992670 to 14155992671. It just doesn't get any simpler than that.

Example sending multiple messages via JSON

```
{  
  "api_key": "my_very_very_secret_api_key12345",  
  "messages": [  
    {  
      "from": "14155992671",  
      "to": "14155992670",  
      "body": "Hello World!"  
    },  
    {  
      "from": "14155992671",  
      "to": "14155992670",  
      "body": "This is my second message where route parameter has  
been set to gold!",  
      "route": "gold"  
    }  
  ]  
}
```

Please check "Message Parameters" section for more info on what parameters are available.

Sending POST (XML) Request

By POST-ing an XML document to our gateway, you can easily send one or more messages in one single request.

In the XML document there must be two nodes in the root - `api_key` and `messages`:

- `api_key` node must contain your API key (which is a string that has the length of 32 characters)
- `messages` node should contain all of your messages (or a single message), each message being a node itself

Please note that it is recommended to wrap message body parameter within CDATA when sending XML request.

Example sending a single message via XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <api_key>my_very_very_secret_api_key12345</api_key>
  <messages>
    <message>
      <from>14155992670</from>
      <to>14155992671</to>
      <body><![CDATA[Hello World!]]></body>
    </message>
  </messages>
</xml>
```

Example sending multiple messages via XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <api_key>my_very_very_secret_api_key12345</api_key>
  <messages>
    <message>
      <from>14155992670</from>
      <to>14155992671</to>
      <body><![CDATA[Hello World!]]></body>
    </message>

    <message>
      <from>14155992670</from>
      <to>14155992671</to>
      <body><![CDATA[This is my second message where route
parameter has been set to gold!]]></body>
      <route>gold</route>
    </message>
  </messages>
</xml>
```

Please check "Message Parameters" section for more info on what parameters are available.

Responses

Each request you make to our HTTP API will return some response to you.

Here we list those responses:

Status code	Message	Description
20	Accepted!	Your request was accepted.
40	Failed to parse [JSON XML]!	JSON object or XML document you're sending could not be parsed.
41	Invalid API Key	API Key you specified is invalid.
42	User not found	There is no user with specified API Key in our database.
43	There are no messages found	We couldn't find any message(s) in your request. Please double-check contents of your request.
44	You cannot send more than X messages at once	There are too many messages in your request to be sent. Please check our per request messages limit.

Example response JSON

```
{"message": "Accepted!", "status": 20}
```

Example response XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<response>  
  <message>Accepted!</message>  
  <status>20</status>  
</response>
```

Message Parameters

Parameter	Description	Example(s)
body	Text of your SMS Use UTF-8 character encoding! If sending GET request, please don't forget to URL encode!	Hello World!
from	Source identifier This must be contained in 11 alphanumeric characters or up to 16 numeric characters.	<ul style="list-style-type: none"> YourName 14155992670 viahub.net (default)
to	Receiver of SMS Number of the message receiver with country code. Please omit the "+" and "00" at the beginning of the string.	14155992671
route	SMS Route Each SMS can be sent via one of the viahub routes. See "SMS Routes" chapter for more info.	Possible values: <ul style="list-style-type: none"> basic (default) silver gold
type	Message type Each SMS can be of various type. If you don't want to send regular SMS but of different type, see "Message types" chapter.	Possible values: <ul style="list-style-type: none"> binary flash
dlr	Delivery response Whether to get a delivery report to the URL specified in the Dashboard. See "Delivery responses" chapter	Possible values: <ul style="list-style-type: none"> 1 0 (default)
id	Message ID Message ID from your system. Used when delivering reports to the Delivery URL you defined	Possible values: <ul style="list-style-type: none"> foo bar baz my_message_123

Setting any parameter for either GET query-string requests or JSON/XML POST requests is very easy. For example, when setting route parameter in your GET request, just append the &route=VALUE.

Setting the same parameter in your XML document (used for POST requests) is as easy as setting a node with the same name (which is `route`) and setting the desired route as the node's value. Take a look at the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <api_key>my_very_very_secret_api_key12345</api_key>
  <messages>
    <message>
      <from>14155992670</from>
      <to>14155992671</to>
      <body><![CDATA[Hello World!]]></body>
      <route>gold</route>
    </message>
  </messages>
</xml>
```

The similar approach is used when creating JSON objects. The parameter name (which is `route`) should be used as a property name when creating a message object. Take a look at the following example:

```
{
  "api_key": "my_very_very_secret_api_key12345",
  "messages": [
    {
      "from": "14155992671",
      "to": "14155992670",
      "body": "Hello World!",
      "route": "gold"
    }
  ]
}
```

Message Types

Binary messages

When sending binary messages, make sure you're sending a message in valid binary format. You have to set parameter `type` to **binary**.

Example sending a vCard

```
424547494E3A56434152440A56455253494F4E3A322E310A4E3A4A6F686E0A54454  
C3B574F524B3B564F4943453A2B313132333435363738390A54454C3B574F524B3B  
4641583A2B313132333435363738390A454D41494C3A6A6F686E40646F652E636F6  
D0A454E443A5643415244
```

Flash messages

When sending flash messages, you have to set parameter `type` to **flash**.

Unicode messages

When sending messages that contain non-GSM alphabet characters, you don't have to specify any extra parameters at all. Our system will automatically detect and take care of it.

Receiving delivery reports

When a message has the `d1r` parameter set to 1 (true), we will try to deliver it to the URL you have specified.

When specifying the URL, the following replacement variables are available:

Variable	Description
<code>%id%</code>	<code>id</code> parameter you submitted with the message. If none, this defaults to our internal message ID
<code>%status%</code>	Delivery status. Check "Delivery responses" chapter for possible values.

Example:

```
http://foo.com/bar?id=%id%&status=%status%
```

Note: if message destination number was invalid, we never send delivery reports, even if the `d1r` parameter was set to true.

SMS Routes

We have three route levels:

- basic
- silver
- gold

Each route level respectively represents a level of reliability.

Sending a message using **gold** route would be the most secure route, as we are sending the message via most reliable providers.

Mobile Number Portability (MNP) Check

We also offer Mobile Number Portability (MNP) checks on our HTTP API. It is pretty simple and straight-forward to do a MNP check.

All you have to do is make a GET request to the `https://gateway.viahub.net/mnp` with the following GET parameters

- `api_key` - your API Key
- `msisdn` - MSISDN is a number uniquely identifying a subscription in a GSM or a UMTS mobile network. Simply put, it is the telephone number to the SIM card in a mobile/cellular phone.

Example MNP check request

```
https://gateway.viahub.net/mnp?  
api_key=my_very_very_very_secret_api_key&msisdn=12025550189
```

If MNP check was successful, you will receive a JSON object in response with three parameters:

- `response` - would be set to `true`
- `mcc` - mobile country code
- `mnc` - mobile national code

If MNP check was not successful, you will receive a JSON object in response with two parameters:

- `response` - would be set to `false`
- `message` - a message containing more info

Example response for successful MNP check

```
{"response":true,"mcc":"219","mnc":"02"}
```

Example response for failed MNP check

```
{"response":false,"message":"Nothing found for this MNP check."}
```