

August 2022

# SMTP API Specification

## V2.5

## Contents

|  |    |
|--|----|
| 1. Overview                                  | 4  |
| 2. Introduction                              | 4  |
| 3. Getting started                           | 4  |
| 4. Basic email structure                     | 5  |
| 4.1 Authentication                           | 5  |
| 4.2. Send a message                          | 6  |
| 4.3 Text parameter versus the data parameter | 7  |
| 5. Getting started                           | 8  |
| 5.1 Table of parameters                      | 8  |
| 5.2 Message parameters in detail             | 11 |
| 6. Batch messaging                           | 21 |
| 7. 8-bit messaging                           | 23 |
| 8. Messaging examples                        | 23 |
| 8.1 Simple examples                          | 23 |
| 8.2 Batch messaging examples                 | 24 |
| 8.3 8-bit SMS examples                       | 25 |
| 9. Appendix A: Error codes                   | 27 |
| 10. Appendix B: Message Statuses             | 28 |
| 11. Terminology                              | 29 |
| 12. Contact details                          | 29 |

## Important Notice

This API is only supported on Clickatell's SMS Gateway.

Note that no new accounts are being set up, except in very specific circumstances.

This document is provided for historical reasons only, and some of the APIs may no longer be available.

## 1. Overview

This technical document is intended for developers who wish to use the Clickatell SMTP API for sending messages and describes the various programming methods and commands used by developers when using this API.

The SMTP API allows you to send SMS messages via email. Your application can then send SMSs by specifying your username, password, and API ID in the body of the email, together with the message you want to send and the list of recipients.

When you sign up for an SMTP account you will be given a username, password and api\_id: keep these at hand. Once you have registered and been activated you will receive 10 free credits with which to test our service. Messages sent with these credits contain a pre-populated message. You can test the API using these credits, and purchase credits to start sending your own, customized messages.

To reduce testing costs, Clickatell offers a test number range. Messages sent to any number on this prefix will only be charged 1/3 of a credit. Use the number 279991xxxx (for South Africa) or 1999xxxxxxx (for the U.S.) where “xxxx” represents any numeric string. Message statuses will be returned.

We will cover the SMTP method in this document. Additional documentation is available for the other methods.

## 2. Introduction

Quite often, the only way of getting a server to send SMS messages via Clickatell’s gateway is via a server-generated email. This API connection facilitates that operation. It supports most of the functionality of the HTTP API version 2.3.0 including 8-bit messaging and templating.

The recommended batch size is 10 000 messages with a maximum of 50 000 messages per email allowed.

**Note: It is important that the ENTIRE document is read before contacting support.**

## 3. Getting started

In order to use the Clickatell gateway you need a Clickatell account and at least one registered connection (API sub-product instance) between your application and our gateway. Each connection method is known as a sub-product (of our API product). You can follow these steps to get started:

### Step 1 – Login to your account

When you have logged in you will be on the Clickatell Central landing page. You will receive 10 free credits which you can use to test the Clickatell Gateway. Please note that for security reasons these 10 credits contain pre-set Clickatell content.

An HTTP API will be added to your account for you. This will allow you to start testing the Clickatell Gateway immediately. You can purchase credits when you are ready to start sending personalized messages.

## Step 2 – Adding an SMTP API to your account

To add a SMTP API to your account, select **APIs** from the main menu and then select **Setup a new API** from the submenu. Click the Add FTP API button on the Setup API page that opens. You can then complete all the required details to configure your API.

Note: For more information on managing your API connections within your Clickatell account see our API guide at <http://www.clickatell.com/help-support/developer-apis/clickatell-api/>

## 4. Basic email structure

To send a message with the Clickatell SMTP API, the system interprets variables that you pass in the body of the email. Each line represents a variable in the form of:

```
variablename:value  
variablename:value  
variablename:value
```

We endeavor to strip any superfluous white space that may occur either side of the colon. However, to avoid any erroneous characters from creeping into the email by mail clients, we suggest that you keep to the format shown above.

Note: We only accept plain-text emails. We do not process HTML-formatted emails. The “Content-Transfer-Encoding” header may only be set to 7-bit which is the default.

The emails themselves should be sent to:

[sms@messaging.clickatell.com](mailto:sms@messaging.clickatell.com)

The subject field of the message is ignored. All the following examples are of the body only, unless otherwise indicated. Please check to determine after how many characters your email client forces word-wrapping (inserts a line-break). Make sure the length of each line is less than this value. We read a maximum of 8192 bytes from each line, so avoid having too many comma-delimited destination addresses. If uncertain, please test before sending large volumes.

### 4.1 Authentication

In order to deliver a message, the system needs to authenticate the request as coming from a valid source. We use a number of parameters to achieve this:

- **api\_id:** This is issued to you when you register for the API product.
- **user:** This is the username of your account.
- **password:** The current password you have set on your account.

You can send multiple emails on the same API account simultaneously.

**Example**

```
api_id:1234
user:xxxxxxxxx
password:xxxxxxxxxx
```

**Note:** To receive a response from the API, use the 'reply:your email address' parameter, and you will receive an API response in the form of an email. See the **reply** parameter for more information.

**4.2. Send a message**

Once you have set up the three authentication variables listed above, there are only another two parameters that are required before you can send an SMS via SMTP.

```
to:xxxxxxxxxxxxxxxxxx
```

This parameter sets the destination address. You may send the same SMS to one or more recipients in a single mail. There are two ways to achieve this: either using comma-delimited destination addresses (ensure that there are no spaces before or after the comma), for example:

```
to:xxxxxxxxxxxxxxxxxx,xxxxxxxxxxxxxxxxxx,xxxxxxxxxxxxxxxxxx,xxxxxxxxxxxxxxxxxx
```

or multiple lines, e.g.:

```
to:xxxxxxxxxxxxxxxxxx
to:xxxxxxxxxxxxxxxxxx
to:xxxxxxxxxxxxxxxxxx
to:xxxxxxxxxxxxxxxxxx
```

The last variable is the text parameter. The text parameter can run over multiple lines. This enables you to insert new lines in your SMS. It also ensures that when your mail client is line wrapping, that the text delivered to the handset is not cut at the line wrap of your email, for example:

| Email       | Resultant SMS |
|-------------|---------------|
| text:Line 1 | Line 1        |
| text:Line 2 | Line 2        |
| text:Line 3 | Line 3        |

**Simple Example:**

api\_id:1234  
user:xxxxxxxxx  
password:xxxxxxxxxx  
to:1234567890,1234567890  
text:This is my first test SMS

**Note:** It is suggested that the maximum length of each text line be less than 40 characters. However, the maximum length is determined by the email client that you use. As such, this could be more or less than 40 characters.

### 4.3 Text parameter versus the data parameter

An alternative to the **text** parameter is the **data** parameter.

This parameter is used to avoid mal-formatting of an SMS, such as when a mail client inserts an unwanted line break in a text string, due to automatic line wrapping. If this parameter is used, any line break automatically inserted by the mail client will be removed when delivering the SMS.

We will also strip out any white space before and after the message text lines, resulting in an unbroken text line in the resultant SMS. Therefore, use the data parameter when sending binary or Unicode messages.

**Example 1:**

| Email sent  | SMS received  |
|---|---|
| api_id:1234<br>user:xxxxxxxxx<br>password:xxxxxxxxxx<br>to:1234567890,1234567890<br>text:Test SMS with the email client inserting a line break here<br>in this sentence<br>text: . This is a new sentence | Test SMS with the email client<br>inserting a line break<br>. This is a new sentence. |

In the above example the email client automatically inserts a line break before the word “here”. When the SMS is received, all text after this line break (for the first instance of the text parameter) is lost. A line break is then inserted at the end and the contents of the second text parameter added.

**Example 2:**

| Email sent   | SMS received   |
|--|--|
| api_id:1234<br>user:xxxxxxxxx<br>password:xxxxxxxxxx<br>to:1234567890,1234567890 | Test SMS with the email client<br>inserting a line break. This is a<br>new sentence. |

|   |  |
|---|--|
| data:Test SMS with the email client inserting a line break<br>here in this sentence<br>data: . This is a new sentence |  |
|---|--|

In the above example the email client automatically inserts a line break before the word “here”. When the SMS is received, all text after the line break for the first data line is lost. The second data line is then appended to the remaining portion of the first data line.

**Note:** Another alternative to the **text** parameter is the **urltext** parameter. This allows you to use a URL encoded string which is useful when sending special characters such as “new line” or Greek text.

## 5. Getting started

### 5.1 Table of parameters

There are a variety of messaging and SMS features supported by the gateway, which can be activated by including a number of additional parameters. These parameters include those in the table below.

| Name                | Parameter name | Short description   | Default value           | Restricted values  |
|---------------------|----------------|---|-------------------------|--|
| API ID              | api_id         | The value for this mandatory parameter can be found logging in online and going to <b>APIs</b> -> <b>Manage APIs</b>  |                         |  |
| Username            | user           | The username you specified.   |                         |  |
| Password            | password       | Your Developers' Central account password.  |                         |  |
| Destination address | to             | The number of the handset to which the message must be delivered. The number should be in international number format.  |                         | No '00' prefix or leading “+” symbol should be used.   |
| Text                | text           | The text content of the message. Note that some characters take up two characters because of GSM encoding standards   |                         | Go to <a href="http://forums.clickatell.com/clickatell">http://forums.clickatell.com/clickatell</a> search for 'Why do some characters take two spaces?' |
| Source address      | from           | The source/sender address that the message will appear to come from also known as “Sender ID”. These must be registered within your online account and approved by us before they may be used. MO numbers | gateway assigned number | A valid international format number between 1 and 16 characters long, or an 11-character alphanumeric string.  |



|                   |             |  |                 |   |
|-------------------|-------------|--|-----------------|---|
|                   |             | rented from us do not require approval.  |                 |   |
| Enable callback   | callback    | Enables you to receive message delivery statuses via an HTTP, SOAP or XML callback which is posted to a URL of yours using the <b>GET</b> or <b>POST</b> method. This is done every time a message status is updated.  | 0               | 0,1,2,3,4,5,6,7 Read detailed description of parameter. |
| Delivery time     | deliv_time  | Delays delivery of SMS to mobile device in minutes relative to the time at which the SMS was received by our gateway. This should be greater than 10 minutes for best effect. Smaller time frames may be delivered too soon.   |                 | The upper limit is 7 days, or 10080 minutes.            |
| Concatenation     | concat      | Specifies the maximum number of message parts available for the message.   | 1               | 1, 2, 3   |
| Maximum credits   | max_credits | Overrides the maximum charge specified online in "profiles". It works within the bounds of the profiles. In other words, a profile must exist for the maximum credit that you set.   | As per profiles | 0.8,1,1.5,2,2.5,3                                       |
| Required features | req_feat    | Allows you to set the features which must be included when a message is sent. If the route does not support the features which you set as 'required' the message will fail.<br><br><b>Note:</b> The use of this parameter could increase the cost per message if a more expensive gateway is used. |                 | Read detailed description of parameter.                 |
| Delivery queue    | queue       | Delivers the message through one of three queues assigned to each client account. Messages in the highest priority queue will be delivered first.  | 3               | 1, 2,3<br>1 is highest priority.                        |

|                    |          |  |                         |   |
|--------------------|----------|--|-------------------------|---|
| Gateway escalation | escalate | Prompts an escalation to an alternative route, if messages are queued on the least-cost route.   | 0                       | 0 - off<br>1 - Escalate immediately to an alternative route if messages are queued on the least-cost route. |
| Mobile originated  | mo       | This is only applicable to clients that have subscribed to a two-way messaging service. We route via a pre-defined carrier to enable the ability for a reply to be received back.  | 0                       | 0 – Off. We use our normal routing rules.<br>1 – Enable Reply.  |
| Client message ID  | cliMsgId | Client message ID defined by user for message tracking.  |                         | Up to 32 alphanumeric characters. No spaces.  |
| Unicode message    | unicode  | Two-digit language code. Convert your text to Unicode [UCS-2 encoding]. See <a href="http://www.Unicode.org/">http://www.Unicode.org/</a> .  | 0                       | 0 – No Unicode<br>1 – Send as Unicode.  |
| Message type       | msg_type | Message types are associated with a structure that defines the fields of the message, e.g., logos and ringtones. See Message Types for more information.   | SMS_TEXT                |   |
| User data header   | udh      | Informs the mobile handset of the type of data and data length of the user data part of an SMS message. The UDH header is used in conjunction with Binary content to define message types. See 8-bit messaging for more information. |                         | Set UDH data manually.  |
| Data               | data     | The data content of a message, if the UDH component is set manually.   |                         |   |
| Validity period    | validity | The validity period in minutes relative to the time at which the SMS was received by our gateway. The message will not be delivered if it is still queued on our gateway after this time.  | 1440 minutes (24 hours) | Set value in X minutes from 1 – 1440 minutes.   |

The following additional parameters are also available to the SMTP API

| Name             | Parameter name | Short description   | Default value | Restricted values   |
|------------------|----------------|---|---------------|---|
| Reply            | reply          | One or more email addresses which will be sent either a unique identifier for each message or an error code   |               |   |
| CSV Line         | csv            | Used in conjunction with text parameter to handle personalized batch messaging  |               | The mobile number must be the first value in the CSV field. |
| Delimiter        | delimiter      | The mobile number must be the first value in the CSV field.   |               |   |
| CSV Template     | csvtemplate    | Used in conjunction with text parameter to handle personalized batch messaging where additional message parameters need to be set on a message-by-message basis |               |   |
| URL encoded text | urltext        | Useful for special Greek characters, etc.   |               |   |

## 5.2 Message parameters in detail

### 5.2.1 Destination address (to)

SMS messages need to be sent in the standard international format, with country code followed by number. No leading zero to the number and no special characters such as "+" or spaces must be used. For example, a number in the UK being 07901231234 should be changed to 447901231234.

If the optional API setting titled 'Replace the leading zero with correct country code' is enabled for the API in your Developers' Central account, any mobile numbers starting with zero will have the zero stripped and replaced with the international dialing code.

#### **Parameter:**

`to:xxxxxxxxxx`

#### **Text**

This is the default parameter that is used to add message content. A single text message can contain up to 160 characters or 140 bytes.

### 5.2.2 Source address (from)

The source address (**from**), also known as the sender ID, can be either a valid international format number between 1 and 16 characters long, or an 11-character alphanumeric string. These must be registered

within your online account and approved by us before they may be used. MO numbers rented from us do not require approval.

Note that characters such as spaces, punctuation, Unicode and other special characters may not always be supported to all destinations and could interfere with your delivery. We suggest that you refrain from **using such characters on the source address**. The use of an alphanumeric source address with 8-bit messaging may cause message failure. This service is not guaranteed across all mobile networks and may interfere with delivery to certain handsets.

**Note:** To ensure that this feature is supported when delivering your message, the required features (**req\_feat**) parameter for this feature must be set.

**Parameter:**

**from:**xxxxxxxxxx

**Note:** To ensure that this feature is supported when delivering your message, the required features (**req\_feat**) parameter for this feature must be set.

### 5.2.3 Delivery acknowledgement (deliv\_ack)

In order to determine whether an SMS has been received by a handset or not, we request delivery acknowledgement for every message we send. The ability to receive reliable delivery acknowledgements varies between mobile networks. Please test to a specific mobile network first, before assuming that you will receive handset acknowledgments for messages that are delivered.

If a GSM handset is 'absent', e.g., switched off or out of coverage, the SMS will be delivered according to a retry cycle once the handset is back in coverage. A delivery receipt will only be returned if and when the retry is delivered. If the validity period or retry cycle (typically 24 hours) is exceeded, the SMS will fail and show 'Error Delivering Message' or status 8.

Delivery acknowledgements can be monitored via the callback system or online reports.

### 5.2.4 Callback System (callback)

Final or intermediary statuses are passed back by the API depending on the callback value set in the original post. This is done by means of:

- HTTP GET
- HTTP POST
- XML GET
- XML POST
- SOAP GET
- SOAP POST

The variables returned are **apiMsgId**, **cliMsgId**, **to**, **timestamp**, **from**, **status** and **charge**.

### Validation of Callback URL

The URL entered in your Clickatell central account to receive 'SMS Status notifications' is validated to check if a callback can be completed. The URL must begin with either http:// (non-encrypted) or https:// (encrypted). If the callback URL is invalid, a message is displayed indicating an Invalid URL.

### Callback retry interval

The MT callback system will make 8 attempts to deliver a callback to your specified callback URL.

For Example:

1. 2 minutes after the original attempt
2. 4 minutes after last retry
3. 8 minutes after last retry
4. 16 minutes after last retry
5. 32 minutes after last retry
6. 64 minutes after last retry
7. 128 minutes after last retry
8. 3 days after last retry (max retries reached)

### Optional Callback username and password

An optional “username” and “password” can be set in the preferences section of your API product. This username and passwords are not the same as your Clickatell username and password but is a setting of your choice to add additional security.

| Callback value | Message status types returned                                    | Message status code returned      |
|----------------|--|-----------------------------------|
| 0              | No message status returned.                                      |                                   |
| 1              | Returns only intermediate statuses.                              | 002, 003, 011                     |
| 2              | Returns only final statuses of a message.                        | 004, 005, 006, 007, 008, 010, 012 |
| 3              | Returns both intermediate and final statuses of a message.       | All except 001                    |
| 4              | Returns only error statuses of a message.                        | 005, 007, 009, 010, and 012       |
| 5              | Returns both intermediate and error statuses of a message.       | 003, 005, 007, 009, 010, 012      |
| 6              | Returns both final and error statuses of a message               | 004, 005, 007, 009, 010, 012      |
| 7              | Returns both intermediate, final and error statuses of a message | 003, 004, 005, 007, 009, 010, 012 |

## Examples

### - HTTP

Sample callback to your callback URL using an HTTP get:

[https://www.yoururl.com/script.asp?api\\_id=12345&apiMsgId=996f364775e24b8432f45d77da8eca47&cliMsgId=abc123&timestamp=1218007814&to=279995631564&from=27833001171&status=003&charge=0.300000](https://www.yoururl.com/script.asp?api_id=12345&apiMsgId=996f364775e24b8432f45d77da8eca47&cliMsgId=abc123&timestamp=1218007814&to=279995631564&from=27833001171&status=003&charge=0.300000)

### - XML

The following data is sent in XML MT callbacks in a parameter called 'data':

```
<?xml version="1.0"?>
<callback>
  <apiMsgId>996411ad91fa211e7d17bc873aa4a41d</apiMsgId>
  <cliMsgId></cliMsgId>
  <timestamp>1218008129</timestamp>
  <to>279995631564</to>
  <from>27833001171</from>
  <charge>0.300000</charge>
  <status>004</status>
</callback>
```

Sample callback to your callback URL using an **XML GET**:

<https://www.yoururl.com/script.php?data=996411ad91fa211e7d17bc873aa4a41d1218008129279995631564278330011710.300000004>

### - SOAP

With the SOAP callback method, a SOAP packet will be sent with a parameter called 'data'. Below is an example packet that will be sent to you via GET or POST.

Example of a SOAP packet that will be sent to you via **GET** or **POST**:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="mt_callback">
  <SOAP-ENV:Body>
    <tns:mt_callback xmlns:tns="mt_callback">
      <api_id xsi:type="xsd:int">1234</api_id>
```

```
<apimsgid xsi:type="xsd:string">2e838df2ee3ea418272ae05aaf84ce5d</apimsgid>
<climsgid xsi:type="xsd:string">abc123</climsgid>
<to xsi:type="xsd:string">27999123456</to>
<from xsi:type="xsd:string">27999000224</from>
<timestamp xsi:type="xsd:int">1213690834</timestamp>
<status xsi:type="xsd:int">003</status>
<charge xsi:type="xsd:float">0.300000</charge>
</tns:mt_callback>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

This is an example callback URL that will be sent to your application:

```
http://www.yoursite.com/your_url.php?data="<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="mt_callback"><SOAP-ENV:Body>
<tns:mt_callback xmlns:tns="mt_callback"><api_id xsi:type="xsd:int">1234</api_id>
<apimsgid xsi:type="xsd:string">2e838df2ee3ea418272ae05aaf84ce5d</apimsgid><climsgid
xsi:type="xsd:string">abc123</climsgid><to xsi:type="xsd:string">27999123456</to>
<from xsi:type="xsd:string">27999000224</from><timestamp
xsi:type="xsd:int">1213690834</timestamp><status xsi:type="xsd:int">003</status>
<charge xsi:type="xsd:float">0.300000</charge></tns:mt_callback></SOAP-ENV:Body></SOAP-
ENV:Envelope>"
```

### 5.2.5 Delivery time (deliv\_time)

The delivery of an SMS message may be delayed by setting an amount of time in **minutes** relative to the time at which it was received by our gateway. We will store the message until the required time frame has elapsed. The maximum delay time is 10080 minutes or 7 days.

#### **Parameter:**

**deliv\_time:120**

### 5.2.6 Concatenation (CONCAT)

If this value is set to 1, 2 or 3 the message will span across 1, 2 or 3 SMS messages where applicable. One text SMS will be sent for every 160 characters or 140 bytes. If a message is concatenated, it reduces the number of characters contained in each message by 3. With 8-bit concatenated messages, each SMS can support up to 140 bytes including the UDH headers.

Please also see [http://support.clickatell.com/faq.php?mode=view\\_entry&kbid=121&kbcac=26](http://support.clickatell.com/faq.php?mode=view_entry&kbid=121&kbcac=26) for information on characters that require two-character places.

Please be aware that a single Unicode SMS can only contain a maximum of 70 characters. You can send 3 Unicode characters less per part when sending a concatenated Unicode message

Values set are:

| Value | Status  |
|-------|---|
| 1     | Default - No concatenation: only 1 message  |
| 2     | Concatenate a maximum of 2 messages.  |
| 3     | Concatenate a maximum of 3 messages.  |
| N     | Concatenate a maximum of N messages. (Delivery is dependent on mobile and gateway. A maximum of 3 is recommended. The maximum number of messages that can be concatenated is 35). |

**Parameter:**

`concat:x`

### 5.2.7 Maximum credits (max\_credits)

This parameter overrides the maximum charge associated with message delivery, as set by the profiles selected within your client account after logging in online. This parameter can be used to limit the cost of a message to a particular value and is bound by the maximum credit value specified in your profiles.

A valid API message ID can still be returned for messages that are not delivered as a result of the maximum credits value set. These messages will have a status of routing error (009).

The credit value in this parameter can be set to any amount of credits. To set your delivery profile, go to **Manage account -> Account overview** and click the link **Control the routing of messages** located in the section titled **Account Type**.

**Parameter:**

`max_credits:3`

### 5.2.8 Required features (req\_feat)

This parameter specifies the features that must be present in order for message delivery to occur. If all features are not present, the message will not be delivered. This prevents SMS messages arriving at a destination via the least-cost gateway, without certain features. This would, for instance, prevent the dropping of a sender ID.

This means that we will not route messages through a gateway that cannot support the required features you have set. For certain message types, we always set the required feature bitmask where relevant. These are FEAT\_8BIT, FEAT\_UDH, FEAT\_UCS2 and FEAT\_CONCAT.

This parameter is set using a combined decimal number to refer to the additional required features.

E.g.: 32 + 512 = 544 – Numeric sender ID and Flash SMS both required.

The value you would set to ensure that Flash and numeric sender ID are both supported, would therefore be **544**.



To ensure that delivery acknowledgment and alphanumeric IDs are supported you would use the value **8240** (16 + 32 + 8192).

| Hex value | Decimal | Feature       | Description                              |
|-----------|---------|---------------|--|
| 0x0001    | 1       | FEAT_TEXT     | Text – set by default.                   |
| 0x0002    | 2       | FEAT_8BIT     | 8-bit messaging – set by default.        |
| 0x0004    | 4       | FEAT_UDH      | UDH (Binary) - set by default.           |
| 0x0008    | 8       | FEAT_UCS2     | UCS2 / Unicode – set by default.         |
| 0x0010    | 16      | FEAT_ALPHA    | Alpha source address (from parameter).   |
| 0x0020    | 32      | FEAT_NUMER    | Numeric source address (from parameter). |
| 0x2000    | 8192    | FEAT_DELIVACK | Delivery acknowledgments.                |
| 0x4000    | 16384   | FEAT_CONCAT   | Concatenation – set by default.          |

**Parameter:**

req\_feat:####

### 5.2.9 Delivery queue

Setting this parameter will assign the message to one of three queues assigned to each user account. This sets the priority of a message sent to us, relative to other messages sent from the same user account. Messages in queue number 1, will always be delivered before messages in queue number 2 and 3, while messages in the 3rd queue, will have the lowest priority (relative to queues 1 and 2).

This is useful when delivering, for example, a single high priority message while you have a large batch going through that same account. The large batch will be queued through queue number 3 (default), and urgent alerts (sent through queue 1), will be delivered ahead of those messages in the batch (queue 3), regardless of when they are actually sent to us.

Values set are:

| Value | Status   |
|-------|--|
| 1     | Use first / primary user queue (highest priority).       |
| 2     | Use second user queue.                                   |
| 3     | Use third user queue (lowest priority) - Default status. |

**Parameter:**

queue:2

### 5.2.10 Gateway escalation (escalate)

By default, the message router will select the lowest cost route (matching features and reliability) that is available for a given destination.

This parameter ensures that, should a message be delayed due to gateway congestion or some other reason on the initial gateway selected by our router, then alternative routes that match the required features will be sought. This is done by moving through the available gateways in order of increasing cost, up to the maximum charge set by the user either using the parameter that defines the maximum credits or based on the profiles selected.

When urgent and high priority messages are sent, they should be posted with escalate set to 1 (on), combined with a high maximum credit value to ensure that the greatest number of gateways are available.

Values set are:

| Value | Status   |
|-------|--|
| 0     | Off – Default value  |
| 1     | On – Escalate immediately to an alternative route if the messages are queued on the least-cost route |

#### **Parameter:**

escalate:1

### 5.2.11 Mobile originated (MO)

This parameter is only used when a message is sent to a handset and a reply is expected.

**PLEASE NOTE: This parameter is only valid for clients that have signed up and paid for our twoway messaging service. An alternative to our least-cost gateway may be used, which could result in a higher cost per message. Please email Clickatell support for pricing or view online.**

When sending a normal MT message to a handset and you expect a reply to your registered MO number, please set the **mo** parameter to “1”.

Values to set are:

| Value | Status   |
|-------|--|
| 0     | Off – Default status. We use the normal routing feature.                                 |
| 1     | Enables reply ability. We route via a predefined carrier to enable the ability to reply. |

It is important that the user specifies the correct from parameter together with this parameter. If no from parameter is specified, we will use a default originator number as set by Clickatell. You will NOT receive these replies.

If you specify the originator (the purchased mo number), then we will route the message such that it can be replied to by the recipient. This reply will be sent to you.

**Parameter:**

mo:1

### 5.2.12 Client message ID (climsgid)

This parameter is set by the user to enable internal message tracking. It allows the user to set their own tracking ID for each message. Once set for a given message, this may be used in place of the Clickatell issued API message ID (apimsgid) for querying message.

A client message ID (**climsgid**) may be any combination of alphanumeric characters excluding spaces. A maximum of 32 characters may be used.

**Parameter:**

climsgid:xxxx

### 5.2.13 Unicode (unicode)

If this value is set to 1, the text field must contain two-byte Unicode. Each SMS can handle a maximum of 70 characters. Each Unicode character must be hex-encoded. More information is available at <http://www.Unicode.org/>.

**Note:** When using the batch send facility for delivering Unicode messages, it is not possible to substitute variables into the message content. This is only possible with Germanic characters.

Values set are:

| Value | Status                                     |
|-------|--|
| 0     | Off – default status                       |
| 1     | On – delivers the text as two-byte Unicode |

We provide a converter to convert text to Unicode within your client account online. Go to “Converters” from within your account online.

**Parameter:**

unicode:1

*E.g.* “ΩΨΘ” becomes

data:03A903A80398

unicode:1

#### 5.2.14 Message types (msg\_type)

A wide variety of messages can be sent through our gateway. We have pre-defined a number of SMS message-types in the API, so that you do not have to set the UDH (user data header) manually. You may optionally set the UDH rather than using one of the message types set below.

For non-Nokia message types (EMS, etc.), please generate your own UDH and data according to the manufacturer's specifications of the message type you wish to send.

**This parameter need not be included if the SMS is a standard text message.**

Values set are:

| Value    | Description   |
|----------|---|
| SMS_TEXT | This is the default message type. It is optional to specify this parameter. |

**Parameter:**

Please see the [messaging examples at the end of this document](#).

#### 5.2.15 Validity period (validity)

A message may be given a time frame for which it is valid. After this period the message will expire. This parameter takes an amount of time in minutes relative to the time at which the message was received by our gateway. If the message is queued on our gateway for a period exceeding the validity period set, then a routing error of 115 will be returned. The default validity period is 1440 minutes (24 hours).

**Note:** The validity period is not passed on to the upstream gateway.

**Parameter:**

validity:120

#### 5.2.16 Reply (reply)

Unique message identifiers (API message IDs) are issued for each message submitted to Clickatell. In order to have API message IDs for each message returned to an email address, set this email address using the reply parameter. Multiple addresses may be comma delimited.

Each message returns a unique identifier in the form of an API Message ID. This can be used to track and monitor any given message. The **apimsgid** is returned after each post. Alternatively, an error code will be returned if the message is not accepted for delivery.

Note: A message status is not returned but may viewed using the online reports or via the HTTP callback.

**Parameter:**

reply:email@yourdomain.com

**Response:**

ID: xxxx To: xxxx

or

ERR: Error number

**5.2.17 URL encoded text (urldata)**

Used instead of the **data** or **text** parameter, the **urldata** parameter can be used to send URL encoded text, which will be decoded back to normal text before the message is delivered to the phone.

It is possible that text could consist of characters that will confuse email-clients. To prevent possible errors, the URL encoding scheme translates all "special" characters to their corresponding hexadecimal codes. These special characters include control characters (carriage returns, line feeds, etc.), certain alphanumeric symbols (% , ' , " , # , & , ? , = , / , :), and other characters (Greek Bulgarian and Cyrillic characters, etc.).

For example, the string "Your URL encoded text!" could be sent as below.

**Parameter:**

urldata:Your%20URL%20encoded%20text%21

**6. Batch messaging**

This facility enables one to do high volume delivery and server-side message merging. It offers the end-user the ability to define all elements common to a batch, and then send only the parameters that change on a message-by-message basis. The following parameters are used for batch messaging.

| Name         | Parameter name | Short description   | Default value | Restricted values   |
|--------------|----------------|---|---------------|---|
| CSV line     | csv            | Each csv line has a mobile number followed by a list of field values that are used to customize a text message. The text parameter value acts as a template with placeholders inserted into the text which are replaced by these field values when the message is sent.         |               | The mobile number must be the first value in the CSV field. |
| CSV template | csvtemplate    | Used in conjunction with the text and csv parameters to handle personalized batch messaging where additional message parameters (such as delivery time) need to be set on a message by message basis. These parameter values are added after the mobile number in the CSV Line. |               |   |
| Delimiter    | delimiter      | Specifies a specific character to be used to delimit values in the CSV  | (comma)       | If you use the default delimiter you will not be            |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | parameter field values. Useful if certain values contain a comma. |  | able to use a comma in your field values. If you are using a TAB character, there must be no white space either before or after the TAB. |
|--|--|---|--|--|

### 6.1.1 CSV line parameter (csv)

`csv:mobile_number[delimiter]field-value[delimiter]field-value[delimiter]field-value` etc.

Where `field-value` is the data to be inserted into the template, and `[delimiter]` is the value as determined by the delimiter parameter.

Placeholders may be inserted within the message body itself. These will take the form of `#field1#` through to `#fieldn#`. See example below. If you wish to customize the whole message on each `csv` line, you would then use:

`text:#field1#`

**Example:**

`api_id:1234`

`user:xxxxxxxx`

`password:xxxxxxxxxx`

`text:Hi #field1#, your balance is #field2#, please come to the office on #field3#`

`delimiter:|`

`csv:1234567890|Fred|€15.50|Mondays,Wednesdays,Fridays`

`csv:1234567890|Jane|€299.45|Tuesdays,Thursdays,Saturdays`

**Note:** If a `,` (comma) was used as the default delimiter, then it would only have passed through `'Mondays'` as the value of field3 for Fred and `'Tuesdays'` for Jane.

### 6.1.2 CSV template parameter (csvtemplate)

`csvtemplate:[parameter-1],[ parameter-2],[parameter-n]`

**Example:** `csvtemplate:deliv_time,cliMsgId,validity` etc.

These values would then appear on the `csv` line immediately after the mobile number. The `field-values` for each of the placeholders that you have inserted in your `text` (text parameter) are then appended to the csv line.

Format: `csv:mobile_number [optional csvtemplate parameters] template_placeholder_parameters`

**Example:**

`api_id:1234`

```
user:xxxxxxxxx
password:xxxxxxxxx
text:Hi #field1#, your balance is #field2#, please come to the office on #field3#
delimiter:|
csvtemplate:deliv_time|cliMsgId|validity
csv:1234567890|10|id_1|60|Fred|€15.50|Mondays,Wednesdays,Fridays
csv:1234567890|40|id_2|60|Jane|€299.45|Tuesdays,Thursdays,Saturdays
```

## 7. 8-bit messaging

Through the SMTP interface, one is also able to send 8-bit messages. These are most often used for ringtones and logos, but one can also send vCards, vCalendar appointments and EMS messages. When sending 8-bit messages, you need to set the user data header (UDH) of the SMS as well as sending the data. If you are comfortable with the creation of your own UDH, we also enable you to set it directly using the **udh** parameter. To simplify the process, we have provided a number of pre-defined message types (see the **msg\_type** parameter).

With the standard **text** parameter, line breaks are automatically inserted. The parameter **data** is thus used for 8-bit messaging.

### Example

```
api_id:1234
user:xxxxxxxxx
password:xxxxxxxxx
to:xxxxxxxxxxxxxxxxx
msg_type:SMS_NOKIA_RINGTONE
data:024A3A5585E195B198040042D9049741A69761781B6176156174288
data:B525D85E0A26C24C49A617628930BB125E055856049865885D200
```

## 8. Messaging examples

### 8.1 Simple examples

#### 8.1.1 Standard email

```
api_id:xxxxx
user:xxxxx
password:xxxxx
to:1234567890
text:This is my first email to SMS message
```

### 8.1.2 Flash SMS with sender ID

```
api_id:xxxxx
user:xxxxx
password:xxxxx
to:1234567890
text:Sending a flash message with sender id.
msg_type:SMS_FLASH
from:ME
```

### 8.1.3 Email with HTTP delivery ack and callback request set

```
api_id:xxxxx
user:xxxxx
password:xxxxx
to: 1234567890
text:Sending a message with delivery acknowledgment.
deliv_ack:1
callback:3
```

## 8.2 Batch messaging examples

### 8.2.1 Sending the same message to multiple recipients

```
api_id:xxxxx
user:xxxxx
password:xxxxx
to:1234567890,9397433991,4387347839
text:This is my first email to SMS message
```

### 8.2.2 Sending a personalized message to multiple recipients

```
api_id:xxxxx
user:xxxxx
password:xxxxx
text:Hi #field1#, your voucher number is #field2#.
Delimiter:|
csv:447901234567|John|agh1234te
csv:447902345678|John|hfe8423ss
csv:447904567890|John|njg6983ju
csv:447903456789|John|cds2267wq
or
text:#field1#
```



Delimiter:;

csv:447901234567,Mary - your appointment is at 9:15 on Saturday 14th May

csv:447902345678,Craig - your appointment is at 14:30 on Wednesday 18th May

### 8.2.3 Configuring individual messages in a batch

api\_id:1234

user:xxxxxxxx

password:xxxxxxxxxx

text:Hi #field1#, your balance is #field2#, please come to the office on #field3#

delimiter:|

csvtemplate:deliv\_time|cliMsgId|validity

csv:1234567890|10|id\_1|60|Fred|€15.50|Mondays,Wednesdays,Fridays

csv:1234567890|40|id\_2|60|Jane|€299.45|Tuesdays,Thursdays,Saturdays

## 8.3 8-bit SMS examples

Note: You cannot set an alphanumeric Sender ID (**from** parameter) when sending 8-bit messages.

### 8.3.1 Sending a ringtone

api\_id:xxxxx

user:xxxxx

password:xxxxx

to:1234567890

msg\_type:SMS\_NOKIA\_RINGTONE

data:024A3A5585E195B198040042D9049741A69761781B61761561728

data:8B525D85E0A26C24C49A617628930BB125E055856049865885D200

### 8.3.2 Sending an operator logo

api\_id:xxxxx

user:xxxxx

password:xxxxx

to:1234567890

msg\_type:SMS\_NOKIA\_OLOGO

data:00480e01ffffffffffffffff80000000000000001800000

data:0000000000018f1b30f1b30f1b30f9f9ff9f9ff9f9ff9f9981999

data:8199981999819f1999f1999f1999f18f9998f9998f9998f9819

data:9981999819998199f9999f9999f9999f98f1998f1998f1998f18

data:000000000000000018000000000000001fffffffffffffff

### 8.3.3 Removing an operator logo

```
api_id:xxxxx
user:xxxxx
password:xxxxx
to:1234567890
msg_type:SMS_NOKIA_CLEAN
data:00
```

### 8.3.4 Sending a VCARD

```
api_id:xxxxx
user:xxxxx
password:xxxxx
to:1234567890
msg_type:SMS_NOKIA_VCARD
urltext:BEGIN%3AVCARD%0D%0AVERSION%3A2.1%0D%0AN%3ABloggs%3BJoe
urltext:%0D%0ATEL%3BPREF%3A%2B1234567890%0D%0AEND%3AVCARD%0D%0A
```

### 8.3.5 Sending a VCAL

```
api_id:xxxxx
user:xxxxx
password:xxxxx
to:1234567890
msg_type:SMS_NOKIA_VCAL
urltext:BEGIN%3AVCALENDAR%0D%0AVERSION%3A1.0%0D%0ABEGIN
urltext:%3AVTODO%0D%0ACATEGORIES%3AMISCELLANEOUS%0D%0A
urltext:SUMMARY%3AMeet+buyers+at+Mario's%0D%0ADTSTART
urltext:%3A20030301T133000%0D%0AEND%3AVTODO%0D%0AEND%3A
urltext :VCALENDAR%0D%0A
```

## 9. Appendix A: Error codes

The following list of error messages are generated by the Clickatell gateway during a validation phase before we accept the message. These error messages are sent back to your application. There will be no message charge if these errors are generated when sending a message. Data regarding messages that do not pass initial validation will not be included in your Clickatell Central reports.

| Number | Description                    | Detail  |
|--------|--------------------------------|---|
| 001    | Authentication failed          | Authentication details are incorrect.   |
| 002    | Unknown username or password   | Authorization error, unknown username or incorrect password   |
| 003    | Session ID expired             | The session ID has expired after a pre-set time of inactivity.  |
| 004    | Account frozen                 |   |
| 005    | Missing session ID             | Missing session ID attribute in request.  |
| 007    | IP Lockdown violation          | You have locked down the API instance to a specific IP address and then sent from an IP address different to the one you set.   |
| 101    | Invalid or missing parameters  | One or more required parameters are missing or invalid  |
| 102    | Invalid user data header       | The format of the user data header is incorrect.  |
| 103    | Unknown API message ID         | The API message ID is unknown. Log in to your API account to check the ID or create a new one.  |
| 104    | Unknown client message ID      | The client ID message that you are querying does not exist.   |
| 105    | Invalid destination address    | The destination address you are attempting to send to is invalid.   |
| 106    | Invalid source address         | The sender address that is specified is incorrect.  |
| 107    | Empty message                  | The message has no content.   |
| 108    | Invalid or missing API ID      | The API message ID is either incorrect or has not been included in the API call.  |
| 109    | Missing message ID             | This can be either a client message ID or API message ID. For example, when using the stop message command.   |
| 110    | Error with email message       |   |
| 111    | Invalid protocol               |   |
| 112    | Invalid message type           |   |
| 113    | Maximum message parts exceeded | The text message component of the message is greater than the permitted 160 characters (70 Unicode characters). Select <b>concat</b> equal to 1,2,3-N to overcome this by splitting the message across multiple messages. |
| 114    | Cannot route message           | This implies that the gateway is not currently routing messages to this network prefix. Please email <a href="mailto:support@clickatell.com">support@clickatell.com</a> with the mobile number in question.               |

|     |   |  |
|-----|---|--|
| 115 | Message expired                                   | Message has expired before we were able to deliver it to the upstream gateway. No charge applies.  |
| 116 | Invalid Unicode data                              | The format of the Unicode data entered is incorrect.   |
| 120 | Invalid delivery time                             | The format of the delivery time entered is incorrect.  |
| 121 | Destination mobile number blocked                 | This number is not allowed to receive messages from us and has been put on our block list.   |
| 122 | Destination mobile opted out                      | The user has opted out and is no longer subscribed to your service.  |
| 123 | Invalid Sender ID                                 | A sender ID needs to be registered and approved before it can be successfully used in message sending.   |
| 128 | Number delisted                                   | This error may be returned when a number has been delisted.  |
| 130 | Maximum MT limit exceeded until <UNIX TIME STAMP> | This error is returned when an account has exceeded the maximum number of MT messages which can be sent daily or monthly. You can send messages again on the date indicated by the UNIX TIMESTAMP. |
| 201 | Invalid batch ID                                  | The batch ID that you have entered for batch messaging is not valid.   |
| 202 | No batch template                                 | The batch template has not been defined for the batch command.   |
| 301 | No credit left                                    | Insufficient credits   |
| 302 | Max allowed credit                                |  |
| 901 | Internal error                                    | Please retry   |

## 10. Appendix B: Message Statuses

These are message statuses that are generated after the Clickatell gateway has accepted the message for delivery. Data regarding messages passing initial validation and accepted for delivery will be included in your Clickatell Central reports.

| Number | Hex   | Description                     | Detail   |
|--------|-------|---------------------------------|--|
| 001    | 0x001 | Message unknown                 | The message ID is incorrect, or reporting is delayed.                                      |
| 002    | 0x002 | Message queued                  | The message could not be delivered and has been queued for attempted redelivery.           |
| 003    | 0x003 | Delivered to gateway            | Delivered to the upstream gateway or network (delivered to the recipient).                 |
| 004    | 0x004 | Received by recipient           | Confirmation of receipt on the handset of the recipient.                                   |
| 005    | 0x005 | Error with message              | There was an error with the message, probably caused by the content of the message itself. |
| 006    | 0x006 | User cancelled message delivery | The message was terminated by a user (stop message command) or by our staff.               |
| 007    | 0x007 | Error delivering message        | An error occurred delivering the message to the handset                                    |
| 008    | 0x008 | OK                              | Message received by gateway.   |

|     |       |                                  |   |
|-----|-------|----------------------------------|---|
| 009 | 0x009 | Routing error                    | An error occurred while attempting to route the message.  |
| 010 | 0x00A | Message expired                  | Message has expired before we were able to deliver it to the upstream gateway. No charge applies.   |
| 011 | 0x00B | Message queue for later delivery | Message has been queued at the gateway for delivery at a later time (delayed delivery).             |
| 012 | 0x00C | Out of credit                    | The message cannot be delivered due to a lack of funds in your account. Please re-purchase credits. |
| 014 | 0x00E | Maximum MT limit exceeded        | The allowable amount for MT messaging has been exceeded.  |

## 11. Terminology

- **Receiving Messages:** A message sent (originating) from a mobile handset to an application via Clickatell.
- **Sending Messages:** A message sent from an application to (terminating on) a mobile handset via Clickatell.
- **Content provider:** This is the Clickatell customer who is offering one or more services that are usually premium rated SMS system.
- **Customer:** A registered Clickatell customer utilizing the Clickatell API for message delivery and receipt.
- **Sender ID:** The “from” address that appears on the user’s handset. This is also known as the message originator or source address. A Sender ID must be registered within your account and approved by us before it may be used.
- **Destination address:** The mobile number/MSISDN of the handset to which the message must be delivered. The number should be in international number format, e.g., country code + local mobile number, excluding the leading zero (0).
- **Source address:** See ‘Sender ID’ above.
- **Short code:** A short number which is common across all the operators for a specific region.
- **Subscriber:** The mobile network subscriber who owns the mobile number (MSISDN) which will send or receive SMSs or be billed for premium rated services.
- **Upstream gateway:** A network operator, third party or our own short message service center (SMSC).

## 12. Contact details

Website: [www.clickatell.com](http://www.clickatell.com)

Help URL: <https://www.clickatell.com/contact-support/>

Support: [support@clickatell.com](mailto:support@clickatell.com)

Sales: [sales@clickatell.com](mailto:sales@clickatell.com)