

---

# **CPA Gateway HTTP v2 Protocol**

(Document version: 4.10, Last Update: November 5, 2015)

<b>1. DOCUMENT VERSION HISTORY .....</b>	<b>4</b>
<i>Document Version 4.10 November 5 2015.....</i>	<i>4</i>
<i>Document Version 4.09 September 3 2013 .....</i>	<i>4</i>
<i>Document Version 4.08 May 11 2011 .....</i>	<i>4</i>
<i>Document Version 4.07 February 22 2011.....</i>	<i>4</i>
<i>Document Version 4.06 January 5 2011 .....</i>	<i>4</i>
<i>Document Version 4.05 March 11 2010 .....</i>	<i>4</i>
<i>Document Version 4.04, November 30 2009 .....</i>	<i>4</i>
<i>Document Version 4.03, June 27 2008.....</i>	<i>4</i>
<i>Document Version 4.02, February 21 2008.....</i>	<i>4</i>
<i>Document Version 4.01, December 14 2007 .....</i>	<i>5</i>
<i>Document Version 4.0, October 31 2007 .....</i>	<i>5</i>
<b>2. INTRODUCTION.....</b>	<b>6</b>
2.1 OVERVIEW .....	6
<b>3. MESSAGES FROM CONTENT PROVIDER TO END USER.....</b>	<b>6</b>
3.1 INTRO.....	6
3.2 URLs FOR THE TELETOPIA INTERACTIVE GATEWAY SERVICE.....	6
<i>URLs for Sending Messages Using HTTP (port 80).....</i>	<i>6</i>
<i>URLs for Sending Messages Using HTTPS (port 443) .....</i>	<i>7</i>
3.3 PARAMETERS.....	8
<i>id.....</i>	<i>8</i>
<i>referenceid.....</i>	<i>8</i>
<i>from .....</i>	<i>8</i>
<i>to.....</i>	<i>8</i>
<i>type .....</i>	<i>8</i>
<i>data.....</i>	<i>9</i>
<i>price.....</i>	<i>9</i>
<i>dcs.....</i>	<i>9</i>
<i>auth.....</i>	<i>9</i>
<i>agelimit.....</i>	<i>9</i>
3.4 RESPONSE VALUES FOR MESSAGES SUBMITTED TO THE TELETOPIA INTERACTIVE HTTP GATEWAY	10
<i>Example of a successfully submitted message.....</i>	<i>10</i>
<i>Example of an unsuccessfully submitted message .....</i>	<i>11</i>
3.5 SECURITY .....	11
3.6 EXAMPLES .....	11
<i>Sending a text SMS in English: .....</i>	<i>11</i>
<i>Sending a text SMS in Hebrew:.....</i>	<i>11</i>
<i>Sending a WAP bookmark: .....</i>	<i>11</i>
<b>4. OVERVIEW FOR RECEIVING MESSAGES .....</b>	<b>12</b>
4.1 INTRO .....	12
4.2 URL .....	12
4.3 THE ACTION PARAMETER .....	12
<b>5. MESSAGES FROM END USER TO CONTENT PROVIDER.....</b>	<b>13</b>
5.1 INTRO .....	13
5.2 PARAMETERS.....	13
<i>action.....</i>	<i>13</i>
<i>id.....</i>	<i>13</i>
<i>from .....</i>	<i>13</i>
<i>to.....</i>	<i>13</i>
<i>encoding.....</i>	<i>13</i>
<i>type .....</i>	<i>13</i>
<i>data.....</i>	<i>14</i>
5.3 HOW MESSAGES ARE DETERMINED TO BE DELIVERED SUCCESSFULLY TO CONTENT PROVIDER .....	14

<b>6.</b>	<b>DELIVERY REPORTS TO CONTENT PROVIDER .....</b>	<b>15</b>
6.1	INTRO .....	15
6.2	PARAMETERS .....	15
	<i>action</i> .....	15
	<i>id</i> .....	15
	<i>msgid</i> .....	15
	<i>status</i> .....	15
	<i>expl</i> .....	15
	<i>encoding</i> .....	15
6.3	STATUS CODES .....	16
6.4	HOW DELIVERY REPORTS ARE DETERMINED TO BE DELIVERED SUCCESSFULLY TO CONTENT PROVIDER .....	16
<b>7.</b>	<b>BARRED NUMBER REQUEST TO CONTENT PROVIDER .....</b>	<b>17</b>
7.1	INTRO .....	17
7.2	PARAMETERS .....	17
	<i>action</i> .....	17
	<i>number</i> .....	17
7.3	HOW BARRED NUMBER REQUESTS ARE DETERMINED TO BE SUCCESSFUL .....	17
<b>8.</b>	<b>EXAMPLES .....</b>	<b>18</b>
8.1	PHP USING FILE_GET_CONTENTS() .....	18
8.2	PHP USING CURL .....	19
8.3	PYTHON .....	20
<b>9.</b>	<b>REFERENCES .....</b>	<b>21</b>

## 1. Document Version History

### Document Version 4.10 November 5 2015

- Removed information about MMS.
- Removed section about security (5.4).
- Removed test requests from action parameter in section 4.3 and removed test request section.

### Document Version 4.09 September 3 2013

- Updated the URLs for the Teletopia Interactive gateway service.
- Clarified section 7.1. The barred message type is only applicable to subscription services.
- Added example code for sending messages in PHP and Python.

### Document Version 4.08 May 11 2011

- Changed information about the allowed path on which outgoing messages from the Content Provider may be delivered (see section 3.2).
- Added clarification about handling the barred request type (see section 7.1). A reply message **MUST NEVER** be sent to the end-user when processing this request type.

### Document Version 4.07 February 22 2011

- Added agelimit parameter for message content (see section 3.3).

### Document Version 4.06 January 5 2011

- Added Teletopia Interactive IP-address 85.252.75.156 to section **Error! Reference source not found.**

### Document Version 4.05 March 11 2010

- Added information about status codes 257, 261, 262, 265 and 266 (see section 6.3).

### Document Version 4.04, November 30 2009

- Added **referenceid** parameter for messages sent from Content Provider to end-user (see section 3.3).

### Document Version 4.03, June 27 2008

- Fixed example code.

**Document Version 4.02, February 21 2008**

- Added barred action type.

**Document Version 4.01, December 14 2007**

- Updated section 3.3: All text messages submitted to the gateway must be UTF-8 encoded.

**Document Version 4.0, October 31 2007**

- Initial version of new protocol specification.

## 2. Introduction

### 2.1 Overview

This document contains information required for Content Providers to send and receive SMS messages to mobile-phone subscribers using the Teletopia Interactive HTTP Gateway interface. The interface is implemented by using standard HTTP GET and POST requests for sending and receiving messages from and to end users.

The Content Provider must specify a HTTP address in the form of a URL (configurable through our service configuration web pages) that will be responsible for receiving messages sent from end-users. Messages being sent from the Content Provider to the end user must be sent by using the URL of the Teletopia Interactive HTTP Gateway (see section 3.2).

The interface is designed in order to allow a simple and robust implementation regardless of which platform the Content Provider decide to implements their solutions.

## 3. Messages *from* Content Provider *to* End User

### 3.1 Intro

This section describes the protocol used by the Content Providers to send SMS messages to mobile-phone subscribers. The messages are sent to a URL with the message content, phone number, price and other information passed as arguments in a HTTP GET or HTTP POST request. In case of HTTP POST, the Content Provider must use “application/x-www-form-urlencoded” content encoding (which is also the default way web-browsers will send data from an HTML-form).

### 3.2 URLs for the Teletopia Interactive Gateway Service

The URLs used by the Content Providers for sending messages to the end user are listed below.

#### **URLs for Sending Messages Using HTTP (port 80)**

Primary URL: <http://api1.teletopiasms.no/httpbridge2/>

Secondary URL: <http://api2.teletopiasms.no/httpbridge2/>

Tertiary URL: <http://api3.teletopiasms.no/httpbridge2/>

### **URLs for Sending Messages Using HTTPS (port 443)**

Primary URL: <https://api1.teletopiasms.no/httpbridge2/>

Secondary URL: <https://api2.teletopiasms.no/httpbridge2/>

Tertiary URL: <https://api3.teletopiasms.no/httpbridge2/>

If there's a problem connecting to one of the URLs, the others may still very well work. We encourage the Content Providers to design their systems to switch URL whenever they lose connection.

### 3.3 Parameters

The parameter names are **case-insensitive** and may be passed either as a HTTP GET or POST request. All values are to be URL-encoded as per RFC1738.

#### **id**

A unique identifier for the message. A good choice of value for this parameter is a timestamp (like number of milliseconds since some point in time) or a counter. This will allow us to trace messages through our system as well as sending back Delivery Reports to the Content Provider. Delivery Reports contains the reference to this id. Request which specify an already submitted id will be rejected by the gateway. Example 'id' value: "1051218018547". The id may contain up to 100 characters.

#### **referenceid**

*Note: Not all CPA agreements between Teletopia Interactive and the Content Provider require this field to be set. However, the field should be set to a valid value when possible.*

This parameter must be set to the message id of the message received from the end-user (see **id** parameter in section 5.2) in order to be able to send a premium-rate reply message. This parameter is used for tying the incoming message with the reply message in order to limit the number of premium-rate messages that may be sent from the Content Provider to the end-user.

#### **from**

Specifies the number to send from. This must be a number Teletopia Interactive has allowed the Content Provider to use. Example 'from' value: "2105".

#### **to**

Specifies the number to send to. This must be complete with country code and a plus sign as the first and only allowable symbol. Example 'to' value: "+4700000000". When sending the same message to multiple recipients, a comma separated list of numbers may be supplied (**Note: Each recipient number will then generate a separate delivery report**).

#### **type**

The type of message. Current supported types are:

##### **text**

A regular text SMS. The **data** parameter contains the actual text encoded in **UTF-8**. If the message is longer than the allowed 160 characters for a single SMS, it will automatically be converted into a number of smaller messages and sent as a concatenated SMS.



**raw**

Raw data sent to the cell phone. See also the DCS parameter. The *data parameter* is encoded as specified in GSM 03.40, section 9.2.3.24 (with the exception noted below) and then transformed into a string containing a hexadecimal representation of the data. Using this message type, you handle everything yourself. Note: The first byte in the data parameter is the UDHL (User Data Header Length), so if you are not using User Data Header, you must set this byte to zero.

**data**

The message data. This is dependant on the **type** described above.

**price**

The amount the end-user will be attempted billed. Example 'price' value: "6.50".

**dcs**

The Data Coding Scheme (See GSM 03.38) (Only applicable when Type=Raw)

**auth**

This is an alternative method of specifying the HTTP Basic Authentication code. If both this parameter and the HTTP Basic Authentication header field are present, the header field will take precedence.

**agelimit**

This parameter specifies the age limit for the content of the message. Example: If **agelimit=18**, the message will be rejected if the recipient is less than 18 years old.

### 3.4 Response values for messages submitted to the Teletopia Interactive HTTP Gateway

Messages submitted to the Teletopia Interactive HTTP Gateway can be rejected for a number of reasons, it is therefore important that the response value for all submitted messages is checked. The response value is found in the content body of the HTTP response. There are only 2 possible response values, these are “**ok**” to indicate success and “**err**” to indicate failure. While the successfully submitted messages will always contain the 2-character string “ok”, a failure will also contain an explanation. The explanation follows the “err” string, for example:

```
err: No authorization code supplied.
```

#### Example of a successfully submitted message

```
client:~$ telnet apil.teletopiasms.no 80 ↵
Trying 195.159.20.32...
Connected to apil.teletopiasms.no.
Escape character is '^]'.
GET
/httpbridge2/?auth=<auth_code>&id=123&from=2105&to=%2b47<number>&type=text&data=Hello%20World&price=0 HTTP/1.1 ↵
Host: apil.teletopiasms.no ↵
↵

HTTP/1.1 200 OK
Content-Type: text/plain; charset=iso-8859-1
Content-Length: 3

ok
Connection closed by foreign host.
client:~$
```

Messages that are successfully submitted will simply return the 2-character string “ok” in the content body.

## Example of an unsuccessfully submitted message

```
client:~$ telnet api1.teletopiasms.no 80 ↵
Trying 195.159.20.32...
Connected to api1.teletopiasms.no.
Escape character is '^]'.
GET
/httpbridge2/?auth=<auth code>&id=123&from=2105&to=%2b47<number>&type=text&da
ta=Hello%20World&price=0 HTTP/1.1 ↵
Host: api1.teletopiasms.no ↵
↵

HTTP/1.1 200 OK
Content-Type: text/plain; charset=iso-8859-1
Content-Length: 58

err: Duplicate ID of earlier message; will not redeliver.
Connection closed by foreign host.
client:~$
```

Notice that the HTTP response code is always 200 for both successfully and unsuccessfully submitted messages.

## 3.5 Security

To be able to use the gateway, you **MUST** supply an authorization code in the HTTP GET/POST requests using either HTTP Basic Authorization (See section 9), or the auth parameter. The authorization code can be found on the service configuration web page.

## 3.6 Examples

Here are a number of examples of how to send various types of messages through the gateway. Be sure to use the correct authorization code for “auth” parameter, and a valid phone number for the “to” parameter.

### **Sending a text SMS in English:**

<http://api1.teletopiasms.no/httpbridge2/?from=2105&to=47xxxxxxx&type=text&price=0&data=Hello%20World&auth=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

### **Sending a text SMS in Hebrew:**

<http://ap1.teletopiasms.no/httpbridge2/?from=2105&to=47xxxxxxx&type=text&price=0&data=%D7%98%D7%99%D7%A4&auth=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>  
[XXX](#)

### **Sending a WAP bookmark:**

<http://api1.teletopiasms.no/httpbridge2/?from=2105&to=47xxxxxxx&type=raw&price=0&dcs=4&data=0B0504C34F000000301020101062D1F2B6170706C696361746966F6E2F782D7761702D70726F762E62726F777365722D626F66B6D61726B730081EA00010045C67F0187151103594D4341000187171103687474703A>

[2F2F7761702E696E706F632E6E6F2F77617064632F626F6F6B6D61726B5F726571756573742E6A73703F64634B65793D66356F4A6F55&auth=xxxxxxxxxxxxxxx](http://2F2F7761702E696E706F632E6E6F2F77617064632F626F6F6B6D61726B5F726571756573742E6A73703F64634B65793D66356F4A6F55&auth=xxxxxxxxxxxxxxx)

<http://api1.teletopiasms.no/httpbridge2/?from=2105&to=47xxxxxxxx&type=raw&price=0&dcs=4&data=0B0504C34F0000003010202656B656D32546A373600010101&auth=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

## 4. Overview for Receiving Messages

### 4.1 Intro

The Teletopia Interactive HTTP Gateway uses **HTTP POST** requests to forward user-originated SMS to Content Providers.

All communication from the Teletopia Interactive HTTP Gateway to the Content Provider, share a single URL.

### 4.2 URL

The Content Provider is responsible for providing the URL used to receive messages and delivery reports sent by the Teletopia Interactive HTTP Gateway. The URL is specified on the service configuration web page for the Content Provider.

### 4.3 The Action Parameter

All requests sent to the URL specified in the service configuration have one common parameter which determines the main purpose of the request, this is the **action** parameter.

See the table below for the various actions defined by the HTTP Gateway.

<b>message</b>	Incoming message (SMS) from end user.
<b>status</b>	Delivery rapport for message sent from Content Provider to end user. This delivery report specifies whether the message was successfully sent to the end user; and whether the end user was successfully billed.
<b>barred</b>	This request is sent to inform the Content Provider that a user has rejected a message, and do not want to receive any further messages from Content Providers. The Content Provider must remove this end-user phone number from any list the user is subscribed to

## 5. Messages *from End User to Content Provider*

### 5.1 Intro

This section describes the protocol used by Teletopia Interactive to forward user-originated SMS to Content Providers. The messages are sent to the URL specified in the service configuration with the actual message content, phone number and other information passed as arguments in a HTTP POST request.

### 5.2 Parameters

The parameter names are **case-insensitive**. All values will be URL-encoded as per RFC1738.

#### **action**

This parameter is always **“message”** for a message request.

#### **id**

Message identifier. This identifier string is guaranteed to be unique, and may contain up to 100 characters.

#### **from**

Specifies the phone number of the end-user this message came from.

#### **to**

Specifies the CPA number this message was sent to.

#### **encoding**

Specifies the character encoding used. Internally, the Gateway uses UTF-8, but you may request to use a different encoding and the Gateway will automatically convert to and from UTF-8 and your desired encoding. This parameter is sent to indicate which character encoding the Content Provider has chosen. The encoding applies for all text fields in the request.

#### **type**

Type of message. Supported type is **“text”**.

#### **text**

A regular text SMS. The *data parameter* contains the actual text encoded in whatever the *encoding parameter* specifies (UTF-8 is default).

**data**

The actual data as specified for the type parameter above.

**5.3 How messages are determined to be delivered successfully to Content Provider**

Messages are determined to be delivered successfully to the Content Provider when the Teletopia Interactive HTTP Gateway is able to connect to the URL specified and the HTTP response code indicates that the HTTP request was successful (HTTP response code 200). If for some reason the Content Provider is able to accept a connection from the Teletopia Interactive HTTP Gateway, but is unable to process the message, then setting the response code to any other value will make the Teletopia Interactive HTTP Gateway try to send the message until it succeeds. The content part of the response may be empty (i.e. "Content-Length: 0").

## 6. Delivery Reports to Content Provider

### 6.1 Intro

This section describes the protocol used by Teletopia Interactive to forward Error- and Delivery Reports from the Gateway to Content Providers. The reports are sent to a URL with the ID of the message it refers to and the actual report passed as arguments in a HTTP POST request. HTTP versions 1.0 and 1.1 are both supported.

### 6.2 Parameters

The parameter names are **case-insensitive**. All values will be URL-encoded as per RFC1738.

#### **action**

This parameter is always **“status”** for a delivery report request.

#### **id**

This is the identifier of a message previously sent from the Content Provider to the Teletopia Interactive HTTP Gateway, and the id of the message for which this status report applies.

#### **msgidx**

If a message has been sent to multiple recipients, each recipient number in the comma separated list will generate one delivery report. Because a message sent to a list of recipients share the same message id, this parameter refers to the index of the recipient number in the list. The first recipient will be **msgidx=1**, the second will be **msgidx=2** etc.

#### **status**

The status code. (See section 6.3)

#### **expl**

An explanation of the status code (See section 6.3). This is just to aid debugging; A Content Provider should always compare against the status code, not the explanation text.

#### **encoding**

Specifies the character encoding used. Internally, the Gateway uses UTF-8, but you may request to use a different encoding and the Gateway will automatically convert to and from UTF-8 and your desired encoding. This parameter is sent to indicate which character encoding the Content Provider has chosen. The encoding applies for all text fields in the request.

### 6.3 Status Codes

The currently defined status codes are:

Code	Explanation
<b>200-249</b>	<b>Positive Delivery Report</b>
201	Message delivered successfully.
<b>250-299</b>	<b>Negative Delivery Report</b>
251	Message not delivered because there was no credit left on the customer's account.
252	Message not delivered because the customer's account has been closed.
253	Message not delivered because the customer is unknown.
254	Message not delivered because of an unknown reason.
255	Message not delivered because the account is barred.
257	Message not delivered because the validity period expired.
261	Message not delivered because the maximum number of messages to this account has been exceeded.
262	Message not delivered because the operator does not support the given message price class.
265	Message not delivered because the account is blocked for cpa. <b>The Content Provider MUST remove the number message was sent to from all subscription services.</b>
266	Message not delivered because the subscriber is not old enough.
<b>500-599</b>	<b>General errors</b>
500	Error in message parameters.
501	Internal error, service entry not found.
550	Unknown error.

### 6.4 How delivery reports are determined to be delivered successfully to Content Provider

Delivery reports are determined to be delivered successfully when the Teletopia Interactive HTTP Gateway is able to connect to the URL specified by the Content Provider and the HTTP response code indicates that the HTTP request was successful (HTTP response code 200). The content part of the response may be empty (i.e. "Content-Length: 0").



## 7. Barred number request *to* Content Provider

### 7.1 Intro

An end-user may be permanently barred from receiving messages sent from Content Providers. This means that the Content Provider must remove this end-user phone number from all lists the user is subscribed to. The Content Provider must not send any further messages to this user unless the user re-subscribes. The barred number request is broadcast to all Content Providers connected to the Teletopia Interactive HTTP Gateway services, it is therefore important that no message is sent to the end-user as a result of this request.

The barred number request is only applicable to subscription services.

**IMPORTANT! The Content Provider MUST NEVER send a reply message to the end-user when handling this request type.**

### 7.2 Parameters

The parameter names are **case-insensitive**. All values will be URL-encoded as per RFC1738.

#### **action**

This parameter is always “**barred**” for a barred number request.

#### **number**

The end-user phone number that the Content Provider must remove from all subscriber lists.

### 7.3 How barred number requests are determined to be successful

Barred requests are determined to be successful when the Teletopia Interactive HTTP Gateway is able to connect to the URL specified by the Content Provider and the HTTP response code indicates that the HTTP request was successful (HTTP response code 200). The content part of the response may be empty (i.e. “Content-Length: 0”).

## 8. Examples

### 8.1 PHP using `file_get_contents()`

Example of sending a message:

```
<?php
$url = 'http://api1.teletopiasms.no/httpbridge2/';

$content = stream_context_create(array(
    'http' => array(
        'method' => 'POST',
        'header' => 'Content-type: application/x-www-form-urlencoded',
        'content' => http_build_query(
            array(
                'auth' => 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx',
                'to' => '47xxxxxxxx',
                'from' => '2105',
                'type' => 'text',
                'data' => utf8_encode('Hello, world! (æøåÆØÅ)'),
                'price' => 0
            )
        ),
        'timeout' => 60
    )
));

$response = file_get_contents($url, FALSE, $content);
print_r($response);
?>
```

## 8.2 PHP using cURL

Example of sending a message:

```
<?php
$url = 'http://api1.teletopiasms.no/httpbridge2/';
$params = array(
    'auth' => 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx',
    'from' => '2105',
    'to' => '47xxxxxxxx',
    'type' => 'text',
    'price' => 0,
    'data' => utf8_encode('Hello, world! (æåEÖÅ)')
);

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($params));
curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, 60);
curl_setopt($ch, CURLOPT_TIMEOUT, 60);

// This should be the default Content-type for POST requests
//curl_setopt($ch, CURLOPT_HTTPHEADER,
//    array("Content-type: application/x-www-form-urlencoded"));

$result = curl_exec($ch);
if(curl_errno($ch) != 0) {
    error_log('cURL error when connecting to '
        . $url
        . ': '
        . curl_error($ch));
}

curl_close($ch);
print_r($result);
?>
```

## 8.3 Python

Example of sending a message:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import urllib, urllib2

url = 'http://apil.teletopiasms.no/httpbridge2/'
values = {
    'auth': 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx',
    'to': '47xxxxxxxx',
    'from': '2105',
    'type': 'text',
    'data': 'Hello, world! (æøåÆØÅ)',
    'price': '0'
}

data = urllib.urlencode(values)
req = urllib2.Request(url, data)
rsp = urllib2.urlopen(req, timeout=60)
content = rsp.read()

print content
```

## 9. References

- RFC 1738 Describes URL encoding. <http://www.rfc-editor.org/rfc/rfc1738.txt>  
RFC 2616 Describes HTTP 1.1. <http://www.rfc-editor.org/rfc/rfc2616.txt>  
GSM 03.38 Describes Data Coding Scheme <http://www.etsi.org>  
GSM 03.40 Describes SMS in detail. <http://www.etsi.org>