



# Outgoing API SMS Server Documentation

## Sending an SMS Text Message using HTTP GET

The TextingForward.com SMS Server exposes an HTTP interface allowing applications with internet connectivity to send SMS text messages.

A request for a page using the structure shown below is all that is needed for you to send an SMS through the TextingForward SMS Server. The endpoint for these HTTP requests are **http://www.textinterface.com/pls/text/api\_tadpole\_sendsms** for HTTP.

Messages should be sent as a HTTP **GET** or **POST** using the parameters listed below. One request should be sent per message.

HTTP/1.1 is enabled so, if sending multiple packets, the TCP/IP connection should be kept open between requests. If sending message with a high transmission rate is required, then several persistent HTTP/1.1 connections should be used concurrently (perhaps 3 or 4). Pipelining requests (ala [Mozilla](#)) is supported but encouraged only where necessary (ie with high message rates and large latency between customer and TextingForward).

## Security Considerations

TextingForward uses multiple components to ensure that only qualified vendors are utilizing our Outbound SMS Servers. All vendors that use the API will be provided with the following:

1. Username
2. Password
3. API Key

In addition, vendors shall also be required to provide TextingForward with the IP address in which they are calling our API. If your IP is not registered and active with our SMS server no request will be processed and you will receive a FAILURE response.

## Parameters for all messages

Several parameters are common to all message types, and should be included in the HTTP request regardless of the specific method being invoked. All parameters in the table below have to be passed to the TextingForward SMS Servers in order to process the request

Please refer to Table 1.1. for a complete list of parameters



# Outgoing API SMS Server Documentation

Name	Description
keyword	Unique Active Keyword assigned to Vendor (use SHORTCODE as Keyword for a DEDICATED shortcode)
usercode	The username account information provided by TextingForward to Vendor
passcode	The password account information provided by TextingForward to Vendor
carriercode	The wireless provider for the mobile number. This is not required if not known.
smsto	This is the mobile number that you are sending a SMS alert to
smsmsg	This is the text message that is being sent to the mobile number (smsto)
apikey	This is the unique key that TextingForward shall provide each valid vendor user
format	TEXT = returns html plain text with a result code XML = return xml document with result code
shortcode	Dedicated Short Code (this is only used for dedicated short code)

**Table 1.1 Required Parameters to call**

The following example shows the common structure of all HTTP requests understood by the TextingForward SMS Server:

```
http://www.textinterface.com/pls/text/api_tadpole_sendsms?keyword=mykeyword
&usercode=myusername&passcode=mypassword&carriercode=VERIZONUS&smsto=14071114
444&.....other parameters...
```

Along with these parameters, there are a variety of other parameters which are needed for the message body, depending on your preference for the encoding/content of the message.

## Parsing responses to GET requests

When a message is successfully received and authenticated by the TextingForward SMS Server, a HTTP success will be returned to the caller (i.e. **200** code) and the HTTP body will contain message identifiers. Additionally if XML is passed for the value of the **format** parameter, then a fully qualified XML document will be returned.

### Text Response example:

\*\*\*\*\*

```
GET /api_tadpole_sendsms?keyword=&usercode=&passcode=&carriercode=&smsto=
&smsmsg=Test HTTP/1.1
Host: www.textinterface.com
```

```
HTTP/1.1 200 OK
Date: Tue, 25 Jun 2002 13:00:20 GMT
Server: Apache/1.3.26 (Unix) mod_ssl/2.8.8 OpenSSL/0.9.6a
Content-Type: text/plain
Transfer-Encoding: chunked
```

**3264662672**



# Outgoing API SMS Server Documentation

## XML Response example:

\*\*\*\*\*

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://www.textinterface.com/">3264662672</string>
```

Both formats will return a result code. If the result code comes back with a number then a successful submission will have occurred as long as a valid carrier code has been chosen. NOTE: Receiving a numeric result code does not mean a guaranteed and successful delivery, but it does signify a successful submission for the carriers to attempt to deliver.

It is important to store the numeric number as this will be important to cross reference when attempting to query the delivery report HTTP service to see the final delivery status that the carriers provide.

When receiving a non-numeric result code, please reference the following table to understand the error at hand:

Result Code	Code Description
INVALID_CARRIER	This code means that you have submitted an incorrect carrier code
INVALID_USER	Signifies the username and/or password submitted is incorrect
INVALID_SERVER	This means that your calling register is not registered with our SMS server
INVALID_APIKEY	The API key is not valid with your account
INVALID_KEYWORD	The Keyword is either not correct or is no longer active
INTERNAL_ERROR	Please email customer support at support@textingforward.com

**Table 1.2 Invalid HTTP request Codes and Reasons**

The following are the VALID Carrier Codes to submit to the TextingForward SMS Servers:

Carrier Name	Carrier Code
ALLTEL	ALLTELUS
AT&T (Former Cingular Customers)	CINGULARUS
SPRINT	SPRINTUS
TMOBILE	TMOBILEUS
VERIZON	VERIZONUS
CINCINATTI BELL	CINEBELLUS
US CELLULAR	USCELLULARUS
DOBSON	DOBSONUS
BOOST	BOOSTUS
VIRGIN	VIRGINUS
AT&T	ATTUS
NextTel	NEXTELUS