



HTTP API VER 1.1

HTTP API for Message Forwarding

(API Version 1.1)

Overview	2
Introduction	2
Messaging	3
Error Codes	4
Message parameter details.....	5-6
Contact Details.....	7



HTTP API VER 1.1

Overview

This document is intended for users, who wish to develop applications using the API SMS Gateway for sending SMS messages. The gateway can be accessed using the HTTP protocol by submitting values by means of the GET method to the API Server.

Introduction

The API Gateway can be accessed in various ways:

- Using HTTP Protocol API and submitting values by the GET & Post method
- Using API.com Web Interface
- Using MS-Excel Plugin
- Using Desktop Application – Dual Messenger

NOTE: Accessing gateway through HTTP protocol is one of the best & the fastest ways to deliver SMS messages.

The API.com gateway HTTP API can be used to send various types of SMS messages including simple text messages, flash message etc. This document gives a detailed explanation of how to use the gateway through the HTTP protocol.

HTTP API VER
1.1



Messaging

For sending any type of SMS message, the gateway requires to be sent various parameters including username & password for authentication purposes. All the parameters need to be sent via HTTP protocol using the GET method. The number of parameters required depends upon the type of SMS message.

API.com has developed a Gateway URL for submitting more than one number at a single hit for faster delivery. You can submit a maximum of 100 Mobile numbers in one hit.

SendMessage API :

Promotional SMS :

<https://api.msggatewayhub.com/smsapi/pushsms.aspx?user=abc&pwd=xyz&to=919898123456&sid=API&msg=test%20message&fl=0>

Transactional SMS :

<https://api.msggatewayhub.com/smsapi/pushsms.aspx?user=abc&pwd=xyz&to=919898123456&sid=API&msg=test%20message&fl=0&gwid=2>

Multiple Messages

<https://api.msggatewayhub.com/smsapi/pushsms.aspx?user=abc&pwd=xyz&to=919898123456,919090909090&sid=API&msg=test%20message&fl=0&gwid=2>

Check Delivery API :

[http://api.msggatewayhub.com/smsapi/checkdelivery.aspx?user=abc&password=xyz
&messageid=messageId](http://api.msggatewayhub.com/smsapi/checkdelivery.aspx?user=abc&password=xyz&messageid=messageId)

Credit Blance API :

Promotional Credit :

<http://api.msggatewayhub.com/smsapi/CheckBalance.aspx?user=abc&password=xyz>

Transactional Credit :

<http://api.msggatewayhub.com/smsapi/CheckBalance.aspx?user=demo&password=demo&gwid=2>



The following are the parameters used:

Parameter name	Parameter Value	Description
User	Abc	Your user name
Password	Xyz	Your Password
Msisdn (to)	919090909090	Destination mobile Number
Sid	Sender ID	Sender ID/From Clause in sms Message.
Msg	Your message	Message to be sent
Fl	0 or 1	Flash message indicator 1 for Yes and 0 for NO
Gwid	2 for Transactional Route	For Transactional Route

Response:

Single Message

Recipient's Mobile number, the date stamp and a unique ID.

The Message Id : 919898123456-200612295008123

Multiple Message

The Message Id : 919898123456-200612295008123

The Message Id : 919898001234-200612295008924

Delivery Report

<https://api.msggatewayhub.com/smsapi/checkdelivery.aspx?user=demo&password=demo&messageid=messageid>

Credit Balance



<https://api.msggatewayhub.com/smsapi/CheckBalance.aspx?user=demo&password=demo&qwid=2>

Error Message

The following are the error messages returned by the API.com gateway in case of Invalid parameters provided.

Failed#Invalid Login

Failed#Unauthorized Access

Failed#Unauthorized IP Address

Failed#Invalid Mobile Number

Failed#Invalid SenderID

Failed#Insufficient Credits

Note: Only the above-defined validations are performed at the server side. All other validations for any parameter are required to be handled at the client side explicitly.

Message parameter details

Username (user)

The username value is passed for authentication purposes. The username is provided to the client when he registers for the HTTP API option from API.com. The username (usr) parameter cannot be more than 20 characters long and can only contain alphabets and numbers, with no spaces or special characters.

Password (password)

The password parameter value is also passed in combination with the username for authentication purposes. The password is provided to the client along with the username when he registers with API.com for sending SMS messages through the Gateway. The password (pass) parameter cannot have more than 20 characters and can contain only alphabets and numbers, with no spaces or special characters.

Mobile Number (msisdn)

The mobile number parameter refers to the destination mobile number to which the message is to be sent. However, the mobile number is required to be in a specific format. Any error in this parameter value would lead to non-delivery of the message. The mobile number should contain only numbers. It must also include the country code at the start beginning. It **should not** contain any symbols (+) or 00 etc. and it must be a valid mobile number supported by the API.com (All Indian Cellular Operators).

Message (msg)

The message parameter refers to the actual message that is to be sent to the destination mobile (**msisdn**). The behavior of the message parameter differs depending on the type of message being sent. The message parameter may consist of up to 160 characters belonging to the following set:

- **A Z a...z 0...9**
- Blank space
- The meta characters \ (line feed)

Due to restrictions in the HTTP protocol format, the following special characters must be encoded in order not to collide with reserved HTTP characters:

Character must be encoded as:

Character	Hexa-decimal Encoded String
&	%26
+	%2B
%	%25
#	%23
=	%3D

For simple text messages, the message can contain numbers, alphabets, spaces and certain special characters. However, the message can contain a maximum of 160 characters, including spaces.

In case of maximum number of characters exceed 160, message is treated as Long Message (Concatenated SMS) and is automatically split in multiple parts based on the length of the SMS. Our gateway can support maximum upto 459 characters of Long Message.

Note: In case of Long Message, credits are deducted accordingly.

Valuable Facts:

1. According to GSM specification, a standard SMS message can contain up to 140 **bytes** of data (payload).

2. GSM encoding represents characters using 7 **bits** instead of 8. This therefore provides a maximum of 160 characters per SMS.

$(140 * 8 \text{ bits}) / 7 \text{ bits} = 160 \text{ Character}$

3. A Concatenated SMS is formed from several standard SMS containing a 7 byte concatenation header (contains the information of various parts of the message) at the beginning of each one. Since this 7 byte header is within the message, it reduces the total size of each SMS to 153 characters each.

Number of SMS	Max length
1	160
2	306
3	459

Sender Id (sid)

The sender Id parameter refers to the message dispatcher or from clause (THE SENDER), from whom the message is sent. The sender Id can either be a valid international number up to sixteen characters long or an eleven character alphanumeric string. The sender Id is an optional clause. In case the sender Id is not specified, the SMSC number is sent as the sender Id to the destination mobile

Sender Id in case of text messaging can only be made up of alphabets & digits. It cannot contain any special characters such as (-,+) etc.

Flash Message (fl)

Flash message parameter indicates whether the message will be a flashed on the destination mobile (**msisdn**). It can contain only the values (0 or 1). Default



value for this parameter is 0. When the flash parameter is sent as 1 (fl=1), the SMS message is flashed on the destination mobile.

Value Description

0 Don't flash message.

1 Flash message on destination mobile screen.

Sample code: Php

```
<?
function PostRequest($url, $referer, $_data) {
// convert variables array to string:
$data = array();
while(list($n,$v) = each($_data)){
$data[] = "$n=$v";
}
$data = implode('&', $data);
// format --> test1=a&test2=b etc.
// parse the given URL
$url = parse_url($url);
if ($url['scheme'] != 'http') {
die('Only HTTP request are supported !');
}
// extract host and path:
$host = $url['host'];
$path = $url['path'];
// open a socket connection on port 80
$fp = fsockopen($host, 80);
// send the request headers:
fputs($fp, "POST $path HTTP/1.1\r\n");
fputs($fp, "Host: $host\r\n");
fputs($fp, "Referer: $referer\r\n");
fputs($fp, "Content-type: application/x-www-form
-urlencoded\r\n");
fputs($fp, "Content-length: ". strlen($data) ."
\r\n");
fputs($fp, "Connection: close\r\n\r\n");
fputs($fp, $data);
$result = "";
while(!feof($fp)) {
// receive the results of the request
$result .= fgets($fp, 128);
}
// close the socket connection:
fclose($fp);
// split the result header from the content
$result = explode("\r\n\r\n", $result, 2);
$header = isset($result[0]) ? $result[0] : "";
$content = isset($result[1]) ? $result[1] : "";
// return as array:
return array($header, $content);
}
$data = array(
'user' => "user",
'password' => "password",
```



```
'msisdn' => "919898123456",  
'sid' => "API",  
'msg' => "Test Message from API",  
'fl' => "0",  
);  
list($header, $content) = PostRequest(  
" https://api.msggatewayhub.com/smsapi/pushsms.aspx",  
// the url to post to  
"http://www.yourdomain.com/sms.php", //  
its your url  
$data  
);  
echo $content;  
?>
```

.....

Sample code: ASP

```
<%  
'Set the values same as you use in your API account  
sUser = "user"  
sPassword ="password"  
'Set the mobile number on which you want to send SMS  
sNumber = "919898123456"  
'Your Message  
sMessage = "Test SMS from API"  
' You approved sender id  
sSenderID = "API"  
Dim objXmlHttpMain , sURL  
sURL=" https://api.msggatewayhub.com/smsapi/pushsms.aspx?user=" & sU  
ser &  
"&password=" & sPassword & "&msisdn=" & sNumber & "&sid=" &  
& sSenderID &  
"&msg=" & sMessage & "&fl=0"  
Set objXmlHttpMain = Server.CreateObject("Msxml2.Serv  
erXMLHTTP")  
on error resume next  
objXmlHttpMain.open "GET",sURL, False  
objXmlHttpMain.setRequestHeader "Content-Type", "tex  
t/urlencoded"  
objXmlHttpMain.send  
'Store response  
response.Write (objXmlHttpMain.responseText)  
%>
```



Sample code: ASP.net (C#)

```
using System
using System.IO
using System.Net
using System.Text
using System.Web
string sUserID = "UserName";
string sPwd = "Password";
string sNumber = "919898123456,919227123456";
string sSID = "API";
string sMessage = "Test SMS From API";
string sURL =" https://api.msggatewayhub.com/smsapi/pushsms.aspx?user="+ sUserID + "&
password=" + sPwd + "&msisdn=" + sNumber + "&sid="
+ sSID + "&msg=" + sMessage
+ "&mt=0&fl=0";
string sResponse = GetResponse(sURL);
Response.Write(sResponse);
public static string GetResponse(string sURL)
{
HttpWebRequest request = (HttpWebRequest)WebRequest
.Create(sURL);
request.MaximumAutomaticRedirections = 4;
request.Credentials = CredentialCache.DefaultCreden
tials;
try
{
HttpWebResponse response = (HttpWebResponse)request
.GetResponse ();
Stream receiveStream = response.GetResponseStream (
);
StreamReader readStream = new StreamReader (receive
Stream, Encoding.UTF8);
string sResponse = readStream.ReadToEnd();
response.Close ();
readStream.Close ();
return sResponse;
}
Catch
{
return "";
}
}
```



VB Dot net.

```
Imports System
Imports System.IO
Imports System.Net
Imports System.Text
Partial Class _Default
Inherits System.Web.UI.Page
Protected Sub Page_Load(ByVal sender As Object,
ByVal e As System.EventArgs)
Handles Me.Load
Dim sURL As String
Dim objReader As StreamReader
sURL
="
https://api.msggatewayhub.com/smsapi/pushsms.aspx?user=abc&pwd=xyz&to=91989
8123456&sid=API&msg=test%20message&fl=0"
Dim sResponse As WebRequest
sResponse = WebRequest.Create(sURL)
Try
Dim objStream As Stream
objStream = sResponse.GetResponse.GetResponseStream()
objReader = New StreamReader(objStream)
Response.Write(objReader.ReadToEnd())
objReader.Close()
Catch ex As Exception
ex.Message
End Try
End Sub
End Class
```

JAVA NEW

```
package javaapplication;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;

public class API {
public static String retval="";

    public static String SMSSender(String user,String password,String
msisdn,String msg,String sid,String fl)
    {
        String rsp="";

        try {
            // Construct The Post Data
```



```
        String data = URLEncoder.encode("user", "UTF-8") + "=" +
URLEncoder.encode(user, "UTF-8");
        data += "&" + URLEncoder.encode("password", "UTF-8") + "="
+ URLEncoder.encode(password, "UTF-8");
        data += "&" + URLEncoder.encode("msisdn", "UTF-8") + "=" +
URLEncoder.encode(msisdn, "UTF-8");
        data += "&" + URLEncoder.encode("msg", "UTF-8") + "=" +
URLEncoder.encode(msg, "UTF-8");
        data += "&" + URLEncoder.encode("sid", "UTF-8") + "=" +
URLEncoder.encode(sid, "UTF-8");
        data += "&" + URLEncoder.encode("fl", "UTF-8") + "=" +
URLEncoder.encode(fl, "UTF-8");

        //Push the HTTP Request
        URL url = new
URL("https://api.msggatewayhub.com/smsapi/pushsms.aspx");
        URLConnection conn = url.openConnection();
        conn.setDoOutput(true);

        OutputStreamWriter wr = new
OutputStreamWriter(conn.getOutputStream());
        wr.write(data);
        wr.flush();

        //Read The Response
        BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        String line;
        while ((line = rd.readLine()) != null) {
            // Process line...
            retval += line;
        }
        wr.close();
        rd.close();

        System.out.println(retval);
        rsp = retval;

    } catch (Exception e) {
        e.printStackTrace();
    }
    return  rsp;
}

public static void main(String[] args) {
    String response = SMSSender("username", "password",
"919898123456", "message", "API", "0");
    System.out.println(response);
}
}
```



**SMMSGATEWAYHUB
Technologies Resources India Pvt. Ltd.**

UG-19 Johri Palace Near Treasure Island Mall MG Road. Indore (M.P.) Pin : 452001

Mr.Abbas (S.M.) : +91 8109146771

Sales : 07313010733/07313010722

Support : **01166477704**

Team,
Support Team!