



Bulk SMS API Manual

Introduction

It is possible to integrate any computer system or any website with our Bulk SMS Gateway. This integration will allow your system or website to send automated SMS messages. The opportunities presented by the Bulk SMS API are limitless. You can send automated SMS alerts, reminders, happy birthday messages, SMS Coupons, promotional alerts, etc. We have developed working sample programming code in the following languages: [Python](#), [C#](#), [PHP](#), [Java](#). These working samples are available in Appendix I of this manual or from this web url: <http://portal.bulksmsweb.com/sample/code.html>

BulkSMSWeb.com provides the most robust and efficient API that can give your customers a perfect experience.

iHubCloud **Bulk** SMS API Manual

www.bulksmsweb.com

WEBSERVICES

This document explains BulkSMS web services protocol.

Access

Web Services URL: <http://portal.bulksmsweb.com/index.php?app=ws>

Parameters

Listed in the below table are webservices parameters for individual tasks.

h	Webservices token. To get the web services token, login to the Bulk SMS system. Got to “My Account”, then Click on “User Configuration”, and for the screen that comes up, copy the Webservices token.
u	Account username
op	pv (for sending SMS)
to	Receipient Mobile Number. You can put multiple number separated by comma
msg	SMS Message
schedule	schedule message delivery, format: YYYY-MM-DD hh:mm:ss

Parameter to can be international formatted mobile number, #groupcode or @username, or a mix of them. Separate by commas for multiple value.

The Sender ID is not supplied. The system will use the Sender ID configured in the system. If you need to use multiple Sender IDs, create a sub-user for each Sender ID and specify the respective username when sending SMS for that Sender ID.

Sending SMS Example:

<http://portal.bulksmsweb.com/index.php?app=ws&u=username&h=48f2f840670d7df13edfe266759b174b&op=pv&to=0772268353&msg=test+only>

Returns JSON data like this:

Successfully Send Message to 263772268353:

```
{ "data": [{"status": "OK", "error": "0", "smslog_id": "93763", "queue": "ab0683efb7a30465ad4da44aec8d2144", "to": "263772268353"}], "error_string": null, "timestamp": 1513172497 }
```

Failed to Send Message:

```
{ "status": "ERR", "error": "100", "error_string": "authentication failed", "timestamp": 1513173399 }
```

Get User credit

Parameters	Name or description
Operation (op)	cr
Mandatory parameter	u h

Retrieve Credit Example:

<http://portal.bulksmsweb.com/index.php?app=ws&u=username&h=48f2f840670d7df13edfe266759b174b&op=cr&to=0772268353&msg=test+only>

Return Codes

Below table listed return coded after unsuccessful call to a webservicess operation. Successful operation will returns an OK data in the response message.

Please note that by default the response message is a JSON encoded message.

Error code	Description
ERR 100	authentication failed
ERR 101	type of action is invalid or unknown
ERR 102	one or more field empty
ERR 103	not enough credit for this operation
ERR 104	webservice token is not available
ERR 105	webservice token not enable for this user
ERR 106	webservice token not allowed from this IP address
ERR 200	send message failed
ERR 201	destination number or message is empty
ERR 400	no delivery status available
ERR 401	no delivery status retrieved and SMS still in queue
ERR 402	no delivery status retrieved and SMS has been processed from queue
ERR 501	no data returned or result is empty
ERR 600	admin level authentication failed
ERR 601	inject message failed
ERR 602	sender id or message is empty
ERR 603	account addition failed due to missing data

ERR 604	fail to add account
ERR 605	account removal failed due to unknown username
ERR 606	fail to remove account
ERR 607	set parent failed due to unknown username
ERR 608	fail to set parent
ERR 609	get parent failed due to unknown username
ERR 610	fail to get parent
ERR 611	account ban failed due to unknown username
ERR 612	fail to ban account
ERR 613	account unban failed due to unknown username
ERR 614	fail to unban account
ERR 615	editing account preferences failed due to missing data
ERR 616	fail to edit account preferences
ERR 617	editing account configuration failed due to missing data
ERR 618	fail to edit account configuration
ERR 619	viewing credit failed due to missing data
ERR 620	fail to view credit
ERR 621	adding credit failed due to missing data
ERR 622	fail to add credit

ERR 623	deducting credit failed due to missing data
ERR 624	fail to deduct credit
ERR 625	setting login key failed due to missing data
ERR 626	fail to set login key

Appendix I

Sample Programming Code

Coding samples below are working out of the box. All the code can be used to send SMS through the Bulk SMS API. It was tested and working perfectly. All you need to do is fill the missing information - **\$username, \$token, \$destinations**. *The mobile numbers in these samples are for Bulk SMS technical team and should be removed when implementing the Web Services in your applications.* The Web-service username and token are case sensitive. You will get the username name when you sign up. You can view the "token" when you login to the bulk sms system. Go to **My Account** on the top Menu and then **User Configuration**. You can integrate this code into the rest of your Applications.

Create your Sender ID before testing. After you login, go to **Settings**, then **Manage Sender ID**. You will not be able to send SMS without a Sender ID. First test your account from our system before using these Code examples!

#Python 3 Sample code

```
from urllib.request import urlopen

from urllib.parse import quote

# BulksMS Webservice username for sending SMS.
# Get it from User Configuration. Its case sensitive.

username = ""

# Webservices token for above Webservice username

token = ""

# BulksMS Webservices URL

bulksms_ws = "http://portal.bulksmsweb.com/index.php?app=ws"

# destination numbers, comma seperated or use #groupcode for sending to group

# destinations = "#devteam,263071077072,26370229338"

# destinations = "26300123123123,26300456456456" #for multiple recipients
```

```
destinations = "263772268353,263732268353,263773098427"

# SMS Message to send
message = "Testing Python Script to BulksMS Webservice"

# send via BulksMS HTTP API
ws_str = bulksms_ws + "&u=" + username + "&h=" + token + "&op=pv"
ws_str = ws_str + "&to=" + quote(destinations) + "&msg=" + quote(message)

http_response = urlopen(ws_str).read()

print(http_response)
```

PHP Sample Code

```
<?php
// suppress error message
error_reporting(0);

// BulksMS Webservice username for sending SMS.
//Get it from User Configuration. Its case sensitive.
$username = '';

// Webservices token for above Webservice username
$token = '';

// BulksMS Webservices URL
$bulksms_ws = 'http://portal.bulksmsweb.com/index.php?app=ws';

// destination numbers, comma seperated or use #groupcode for sending to group
// $destinations = '#devteam,263071077072,26370229338';
// $destinations = '26300123123123,26300456456456'; for multiple recipients
$destinations = '263772268353,263732268353,263773098427';

// SMS Message to send
$message = "Testing PHP Script to BulksMS Webservice";

// send via BulksMS HTTP API
```

```
$ws_str = $bulksms_ws . '&u=' . $username . '&h=' . $token . '&op=pv';  
  
$ws_str .= '&to=' . urlencode($destinations) . '&msg=' . urlencode($message);  
  
$ws_response = @file_get_contents($ws_str);  
  
echo $ws_response;  
  
?>
```

C# Sample Code

```
using System;  
  
using System.Net.Http;  
  
using System.Threading.Tasks;  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        Task T = new Task(ApiCall);  
  
        T.Start();  
  
        Console.WriteLine("Json data.....");  
  
        Console.ReadLine();  
    }  
  
    static async void ApiCall()  
    {  
  
        using (var client = new HttpClient())  
        {  
  
            string username = "";  
  
  
            // Webservices token for above Webservice username  
  
            string token = "";
```

```

// BulkSMS Webservice URL

string bulksms_ws = "http://portal.bulksmsweb.com/index.php?app=ws";

// destination numbers, comma seperated or use #groupcode for sending to group
// $destinations = '#devteam,263071077072,26370229338';
// $destinations = '26300123123123,26300456456456'; for multiple recipients

string destinations = "263772268353,263732268353,263773098427";

// SMS Message to send

string message = "Testing C# to BulkSMS Webservice";

// send via BulkSMS HTTP API

string ws_str = bulksms_ws + "&u=" + username + "&h=" + token + "&op=pv";

ws_str += "&to=" + Uri.EscapeDataString(destinations) + "&msg=" +
Uri.EscapeDataString(message);

HttpResponseMessage response = await client.GetAsync(ws_str);

response.EnsureSuccessStatusCode();

using (HttpContent content = response.Content)
{
    string responseBody = await response.Content.ReadAsStringAsync();

    Console.WriteLine(responseBody + ".....");
}

}

}

}

```

Java Sample Code

```
package javasms;

import java.io.*;

import java.net.*;

public class JavaSMS {

    /**
     * @param args the command line arguments
     */
    public static void main (String[] args) {

        //-----//
        // Step 1: Start creating a few objects we'll need.
        //-----//

        URL url;

        String username,token,bulksms_ws,strJson;

        String destinations,ws_str,message;

        //-----//

        // Step 2: Create and encode our Webservice URL.
        //-----//

        //Username that you use to login to http://portal.bulksmsweb.com
        username = "";

        // Webservices token for above Webservice username.
        //login to http://portal.bulksmsweb.com and got to My Account -> User Configuration
        //to get your token
```

```

token = "";

// BulkSMS Webservices URL
bulksms_ws = "http://portal.bulksmsweb.com/index.php?app=ws";

// destination numbers, or use #groupcode for sending to group
// $destinations = '#devteam'
// $destinations = '26300123123123'

destinations = "263772268353"; //multiple numbers allowed. Separate with commas

// SMS Message to send
message = "Testing Java to BulkSMS Webservice";

// send via BulkSMS HTTP API
try {
    ws_str = bulksms_ws + "&u=" + username + "&h=" + token + "&op=pv";
    ws_str = ws_str + "&to=" + URLEncoder.encode(destinations, "UTF-8") + "&msg=" +
URLEncoder.encode(message, "UTF-8");

    url = new URL(ws_str);

    //-----//
    // Step 3: Open a URLConnection to the url. //
    //-----//

    URLConnection conn = url.openConnection(); // throws an IOException

    //-----//
    // Step 4: //
    //-----//

    // Convert the InputStreamReader to a Buffered Reader. //
    // Buffering the stream makes the reading faster; the //

```

```

// readLine() method of the BufferedReader makes the reading //
// easier. //
//-----//

BufferedReader br = new BufferedReader(
                                new InputStreamReader(conn.getInputStream()));

//-----//
// Step 5: //
//-----//
// Store the whole output in a variable. The output will be //
// JSON string variable. //
//-----//

String inputLine;

strJson = "";

while ((inputLine = br.readLine()) != null) {
    strJson += inputLine;
}

br.close();

//Variable strJson holds the reponse from the Webservice

System.out.println(strJson);

} catch (MalformedURLException mue) {

    System.out.println("Ouch - a MalformedURLException happened.");
    mue.printStackTrace();

    System.exit(1);

} catch (IOException ioe) {

```

```
        System.out.println("Oops- an IOException happened.");  
        ioe.printStackTrace();  
        System.exit(1);  
    }  
  
    }  
  
}
```