Text Marketer Developer Documentation - Wiki Home

Welcome to the Text Marketer wiki. Here you will find all the information you need on how to use our SMS applications.

Here you will find:

- A simple and concrete explanation of how to use our API'S
- Lots of code example in several languages.
- Wrapper's packages for download.
- A full and detailed explanation of how our Rest API works.
- All our Developers Documentation.

You can download any of the information you need in a PDF. To do this just simply go to the 'tools' button in the top right hand corner and choose Export to PDF. If you want to download more than one page, select 'Browse' on the top bar and choose 'Space Operations' and on the left you will have your Export options.

Want to start instantly?

Just look at these code examples below and copy and paste them in to be sending SMS instantly, if you don't have an account you can register **here** and get **10 free text credits**.

PHPJAVAC#/.NETC++/CLICVB.NETASPRuby

```
<?php
    // Simple SMS send function
    function sendSMS($username, $password, $to, $message, $originator) {
        $URL =
'https://api.textmarketer.co.uk/gateway/'."?username=$username&password=$password&op
tion=xml";
        $URL .=
"&to=$to&message=".urlencode($message).'&orig='.urlencode($originator);
        $fp = fopen($URL, 'r');
        return fread($fp, 1024);
    }
    // Example of use
    $response = sendSMS('myUsername', 'myPassword', '4477777777', 'My test message',
'TextMessage');
    echo $response;?>
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;

// Simple send SMS programm
public class SendSMS {
    public static String sendSMS(String username, String password, String to, String
message, String originator) {
        String url;
        StringBuilder inBuffer = new StringBuilder();
        try {
            url = "https://api.textmarketer.co.uk/gateway/" +
                "?username=" + username + "&password=" + password + "&option=xml" +
                "&to=" + to + "&message=" + URLEncoder.encode(message, "UTF-8") +
                "&orig=" + URLEncoder.encode(originator, "UTF-8");
        } catch (UnsupportedEncodingException e) {
            return null;
        }
        try {
            URL tmUrl = new URL(url);
            URLConnection tmConnection = tmUrl.openConnection();
            tmConnection.setDoInput(true);
            BufferedReader in = new BufferedReader(new
InputStreamReader(tmConnection.getInputStream()));
            String inputLine;
            while ((inputLine = in.readLine()) != null)
                inBuffer.append(inputLine);
            in.close();
        } catch (IOException e) {
            return null;
        }
        return inBuffer.toString();
    }
    public static void main(String[] args) {
        // Example of use
        String response = sendSMS("myUsername", "myPassword", "4477777777", "My test
message", "TextMessage");
        System.out.println(response);
    }
}
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace SendSMS {
    class Program {
        public static string SendSMS(string username, string password, string to,
string message, string originator) {
            StringBuilder sb  = new StringBuilder();
            byte[]        buf = new byte[1024];
            string url = "https://api.textmarketer.co.uk/gateway/" +
                "?username=" + username + "&password=" + password + "&option=xml" +
                "&to=" + to + "&message=" + HttpUtility.UrlEncode(message) +
                "&orig=" + HttpUtility.UrlEncode(originator);
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            Stream resStream = response.GetResponseStream();
            string tempString = null;
            int count = 0;
            do {
                count = resStream.Read(buf, 0, buf.Length);
                if (count != 0) {
                    tempString = Encoding.ASCII.GetString(buf, 0, count);
                    sb.Append(tempString);
                }
            }
            while (count > 0);
            return sb.ToString();
        }
        static void Main(string[] args) {
            string respXML = SendSMS("myUsername", "myPassword", "4477777777", "My
test message", "TextMessage");
            Console.WriteLine(respXML);
       }
    }
}
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace SendSMS {
    class Program {
        public static string SendSMS(string username, string password, string to,
string message, string originator) {
            StringBuilder sb  = new StringBuilder();
            byte[]        buf = new byte[1024];
            string url = "https://api.textmarketer.co.uk/gateway/" +
                "?username=" + username + "&password=" + password + "&option=xml" +
                "&to=" + to + "&message=" + HttpUtility.UrlEncode(message) +
                "&orig=" + HttpUtility.UrlEncode(originator);
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            Stream resStream = response.GetResponseStream();
            string tempString = null;
            int count = 0;
            do {
                count = resStream.Read(buf, 0, buf.Length);
                if (count != 0) {
                    tempString = Encoding.ASCII.GetString(buf, 0, count);
                    sb.Append(tempString);
                }
            }
            while (count > 0);
            return sb.ToString();
        }
        static void Main(string[] args) {
            string respXML = SendSMS("myUsername", "myPassword", "4477777777", "My
test message", "TextMessage");
            Console.WriteLine(respXML);
       }
    }
}
```

```
/* * Send SMS C/C++ example need curllib download from http://curl.haxx.se/ */

#include <stdio.h>
#include <tchar.h>
#include <string.h>
#include <curl/curl.h>
#define URLSIZE 256

struct MemoryStruct {
        char *memory;
        size_t size;
};

/* Converts a hex character to its integer value */

char from_hex(char ch) {
        return isdigit(ch) ? ch - '0' : tolower(ch) - 'a' + 10;
```

```
}

/* Converts an integer value to its hex character*/

char to_hex(char code) {
        static char hex[] = "0123456789abcdef";
        return hex[code & 15];
}

/* Returns a url-encoded version of str */

char *url_encode(char *str) {
        char *pstr = str, *buf = (char *)malloc(strlen(str) * 3 + 1), *pbuf = buf;
        while (*pstr) {
                if (isalnum(*pstr) || *pstr == '-' || *pstr == '_' || *pstr == '.'
|| *pstr == '~')
                        *pbuf++ = *pstr;
                else if (*pstr == ' ')
                        *pbuf++ = '+';
                else
                        *pbuf++ = '%', *pbuf++ = to_hex(*pstr >> 4), *pbuf++ =
to_hex(*pstr & 15);
                pstr++;
        }
        *pbuf = '\0';
        return buf;
}

/* CURL Callback write function */

static size_t WriteMemoryCallback(void *contents, size_t size, size_t nmemb, void
*userp) {
        size_t realsize = size * nmemb;
        struct MemoryStruct *mem = (struct MemoryStruct *)userp;
        mem->memory = (char *)realloc(mem->memory, mem->size + realsize + 1);
        if (mem->memory == NULL) {
                /* out of memory! */
                printf("not enough memory (realloc returned NULL)\n");
                exit(EXIT_FAILURE);
        }
        memcpy(&(mem->memory[mem->size]), contents, realsize);
        mem->size += realsize;
        mem->memory[mem->size] = 0;
        return realsize;
}

/* Send SMS */

char * sendSMS(const char *username, const char *password, const char *to, char
*message, char *originator) {
        static char url[URLSIZE] =
"https://api.textmarketer.co.uk/gateway/?option=xml";
        char *encoded;
        CURL *curl;
        CURLcode res;
        struct MemoryStruct chunk;
        chunk.memory = (char *)malloc(1);  /* will be grown as needed by the realloc
above */
        chunk.size = 0;    /* no data at this point */
```

```
        curl = curl_easy_init();
        if(curl) {
                strcat_s(url, URLSIZE, "&username=");
                strcat_s(url, URLSIZE, username);
                strcat_s(url, URLSIZE, "&password=");
                strcat_s(url, URLSIZE, password);
                strcat_s(url, URLSIZE, "&to=");
                strcat_s(url, URLSIZE, to);
                strcat_s(url, URLSIZE, "&message=");
                encoded = url_encode(message);
                strcat_s(url, URLSIZE, encoded);
                free(encoded);
                encoded = url_encode(originator);
                strcat_s(url, URLSIZE, "&orig=");
                strcat_s(url, URLSIZE, encoded);
                free(encoded);
                curl_easy_setopt(curl, CURLOPT_URL, url);
                /* send all data to this function  */
                curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, WriteMemoryCallback);
                /* we pass our 'chunk' struct to the callback function */
                curl_easy_setopt(curl, CURLOPT_WRITEDATA, (void *)&chunk);
                if((res = curl_easy_perform(curl)) != CURLE_OK) {
                        return NULL;
                }
                curl_easy_cleanup(curl);
        }
        return chunk.memory;
}

int main(void) {
        char *response;
        response = sendSMS("myuser", "mypass", "4477777777", "test test test",
"me");
        if(response != NULL) {
                printf(response);
                free(response);
        }
```

```
        getchar();
        return 0;
}
```

```
Imports System.Web

Module Module1
    Public Function SendSMS(ByVal username As String, ByVal password As String,
ByVal toPhone As String,
                            ByVal message As String, ByVal originator As String)
        Dim webClient As New System.Net.WebClient
        Dim url As String = "https://api.textmarketer.co.uk/gateway/?username=" &
                username & "&password=" & password & "&option=xml" &
                "&to=" & toPhone & "&message=" &
System.Web.HttpUtility.UrlEncode(message) &
                "&orig=" & System.Web.HttpUtility.urlencode(originator)
        Dim result As String = webClient.DownloadString(url)
        SendSMS = result
    End Function

    Sub Main()
        Dim result As String = SendSMS("myUsername", "myPassword", "4477777777", "My
test message", "TextMessage")
        Console.WriteLine(result)
        Console.ReadKey()
    End Sub

End Module
```

```
<%@language=JScript%><%
    username = "myUsername";
    password = "myPassword";
    to = "4477777777";
    message = Server.URLEncode("My test message");
    originator  = Server.URLEncode("TextMessage");
    url = "https://api.textmarketer.co.uk/gateway/?&option=xml" +
            "&username=" + username + "&password=" + password +
            "&to=" + to + "&message=" + message +
            "&orig=" + originator;
    var objSrvHTTP;
    objSrvHTTP = Server.CreateObject("Msxml2.ServerXMLHTTP");
    objSrvHTTP.open(url, false);
    objSrvHTTP.send();
    Response.ContentType = "text/xml";
    xmlResp = objSrvHTTP.responseXML.xml;
    Response.Write(xmlResp);%>
```

```
require 'net/http'
require 'uri'

def send_sms(username, password, to, message, originator)
  requested_url = 'https://api.textmarketer.co.uk/gateway/?option=xml' +
                "&username=" + username + "&password=" + password +
                "&to=" + to + "&message=" + URI.escape(message) +
                "&orig=" + URI.escape(originator)
  url = URI.parse(requested_url)
  full_path = (url.query.blank?) ? url.path : "#{url.path}?#{url.query}"
  the_request = Net::HTTP::Get.new(full_path)
  the_response = Net::HTTP.start(url.host, url.port) { |http|
    http.request(the_request)
  }
  raise "Response was not 200, response was #{the_response.code}" if
the_response.code != "200"
  return the_response.bodyend
resp = send_sms('myUsername', 'myPassword', '4477777777', 'My test message',
'TextMessage')

puts(resp)
```

If you don't know your API username and password, you'll find them when you log on to the Text Marketer web interface here:

https://messagebox.textmarketer.co.uk/(Account Settings > API Settings)

Need something else?

Please don't hesitate to contact us.

We are here to help.

Related Content

- Sending SMS
- Handling Incoming SMS
- The Text Marketer Email-to-SMS Gateway
- Simple Connectivity with Wrappers
- API Documentation
- All Documentation