

A Quick Guide to the C# SMS REST Client

Integrating the Text Marketer API with your code

Integration Guide

Version 1.2

April 2012

www.textmarketer.co.uk

Contents

Document Change Log.....3

 Introduction.....4

Getting Started.....4

 A Basic Command.....5

Testing with the Sandbox.....6

Functions Available.....7

Document Change Log

Date	Version	Detail
03 March 2012	1.0	First publication
12 April 2012	1.1	Minor bug fix
18 April 2012	1.2	Added sub-account creation function

Introduction

The REST API is a powerful tool for programmatically controlling your Text Marketer account. It is the most feature-rich of all our APIs and provides the most detailed information about the success/failure of your requests.

This document describes the C# REST client library, which makes it really easy to use the REST API in your C# code by just using the provided C# source code file and making the appropriate function calls.

For more advanced use of, and errors returned by, the REST API please see the document **The Complete Text Marketer RESTful SMS Services API** also available from www.textmarketer.co.uk/developers/.

Getting Started

You will need a copy of the C# source code library RestClient.cs or the DLL RestClient.dll available from www.textmarketer.co.uk/developers/ (go to the *Available Libraries* section of the REST API page).

You must use the RestClient assembly in your C# code:

```
using RestAPI;
```

References needed by C# RestClient library :

```
Microsoft.CSharp
System
System.Core
System.Data
System.Data.SetExtensions
System.Web
System.Xml
System.Xml.Linq
```

A Basic Command

The following is a complete example of using the REST client library. The C# REST client may throw exceptions, which means that you need to place any function calls in a try-catch block, as follows:

```
using System;
using RestAPI;
using System.Collections;

namespace ExampleRestAPI
{
    class Example
    {
        static void Main(string[] args)
        {
            RestClient rClient = new RestClient("myusername",
"mypassword", RestClient.ENV_SANDBOX);
            try
            {
                if(rClient.isLoginValid())
                    Console.WriteLine("Login is OK!");

                int credits = rClient.getCredits();
                Console.WriteLine("Account have {0} credits.", credits);

                Hashtable result = rClient.sendSMS("Hello SMS World!",
"447777123123", "Hello World", 72, "", "");
                Console.WriteLine("Used {0} Credits, ID:{1}, Status: {2}",
result["credits_used"], result["message_id"], result["status"]);
            }
            catch (RestClientException e)
            {
                Console.WriteLine(e.Message);
                foreach (DictionaryEntry de in rClient.getLastErrors())
                    Console.WriteLine("Error {0}: {1}", de.Key, de.Value);
            }

            Console.WriteLine();
            Console.WriteLine("Press ENTER to continue");
            Console.ReadLine();
        }
    }
}
```

That's it! Obviously you would need to change the code to use your own API username/password. You can find your API username and password (which may be different to your web interface username/password) via the web interface:

<http://apps.textmarketer.co.uk/account/> (Settings > API Settings)

If you don't have an account, you can set one up for free at www.textmarketer.co.uk.

Testing with the Sandbox

A sandbox system is available to you for testing. The sandbox will not modify your account in any way, but will respond normally to your requests. In other words if you use the sandbox to get the number of credits on your account, it will provide the correct number. However if you use it to transfer credits between accounts, it will simulate a successful transfer (assuming you have enough credits on the account, and provide valid target account details, etc.) but it will not actually modify the number of credits on either account.

Similarly, the sandbox system will not actually send SMS messages and will not deduct credits for simulated sends.

To use the sandbox simply specify the sandbox when you create the client object:

```
RestClient rClient = new RestClient("myusername", "mypassword",  
RestClient.ENV_SANDBOX);
```

Functions Available

When you download the C# library (see *Getting Started* above) you will find automatically generated documentation included. This is the best place to see what functions are available and what arguments they take. You can also read it online here:

www.textmarketer.co.uk/developers/documents/csharp_client/.

A simplified list of RestClient class methods is given below.

- addGroup – create a new 'send group'
- addNumbersToGroup – add new numbers to a 'send group'
- createSubAccount – create a new sub account
- getCredits – get the number of credits available in your account
- getDeliveryReport – get the contents of a delivery report
- getDeliveryReports – get a list of the available delivery reports
- getGroup – get the numbers in a group
- getGroups – get a list of the available 'send' and 'merge' groups
- getKeyword – get the availability of a keyword for use on our 88802 short code number
- getLastErrors – get the last errors returned from the last function call (to the API)
- getLastErrorCode – get the last error code raised from the last RestClient call
- getLastErrorMessage – get the last error message raised from the last RestClient call
- Xml – get the xml string returned from the last RestClient call
- isLoginValid – check the username/password credentials used are valid
- sendSMS – send an SMS to a given recipient
- transferCreditsToAccount – transfer credits to a specified account number
- transferCreditsToUser – transfer credits to a specified account username