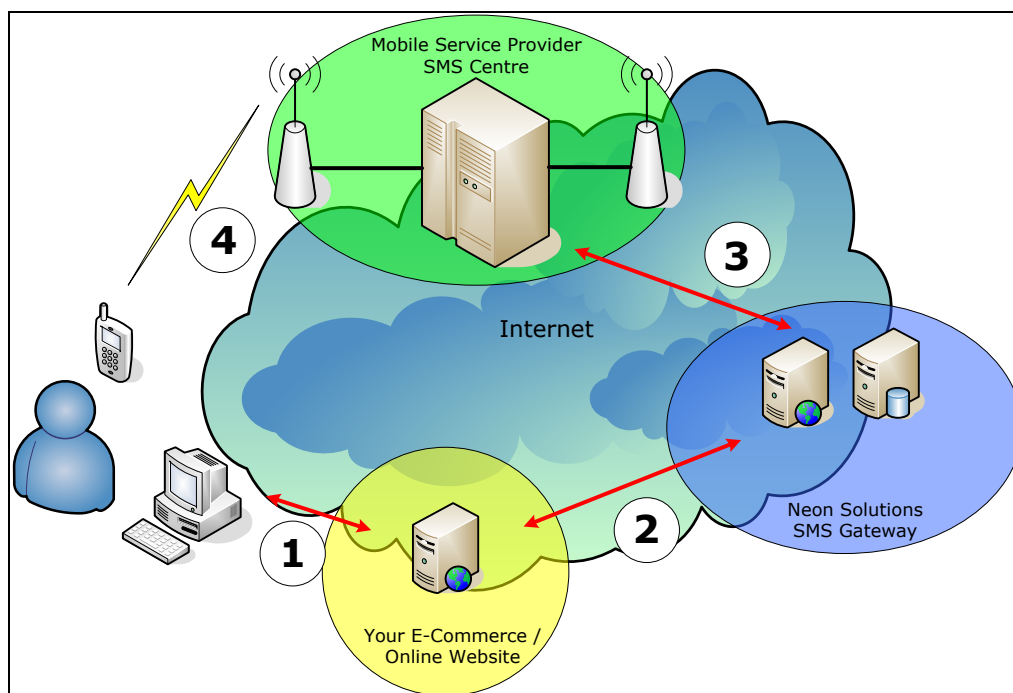## Introduction:

Neon Solutions HTTP/S interface is simple to use, yet makes use of a highly efficient lightweight protocol – HTTP. Our application programming interface accepts both standard and SSL connections so you can be confident that all communications with our SMS Gateway servers is secure and confidential. This document has five different methods described:

- **Basic API Requirements**
- **PHP - Basic HTTP API**
- **PHP - Advanced HTTP API**
- **C# - Advanced HTTP API**
- **VBA - Basic HTTP API**

The five methods are examples of scripting that you can use to interlink with our systems, if you are working with JAVA, .Net or any other dynamic HTML language, our developers can liase with you to ensure complete compatibility with our systems. Using open standards in the design of our systems has been a priority which ensures that by default we can achieve maximum compatibility.

How it works:

Example : Neon Solutions E-Commerce Solutions

1. User logs onto your website on their computer and carries out an online transaction.

2. Your E-Commerce server processes the transactions as it normally would, but then generates a message with your defined information to the Neon Solutions SMS Gateway servers.

3. Neon Solutions SMS Gateway servers take this entry on the website and convert it into a signal to the SMS service centres which control access to the mobile phone transmission network.

4. The SMS service centre receives the message from Neon Solutions SMS Gateway and directs the information to the nearest mobile phone base station and which broadcast to your customer's mobile phone. Once delivered a report is

returned to the Neon Solutions SMS Gateway and logged on your SMS history.

Basic API Requirements:

Simply send a GET or POST request as follows:

https://api.neonsolutions.ie/sms.php?user=<username>&clipwd=<password>&text=<number>&to=<recipient>

Required Variables:

- <u>user:</u> Username
- <u>clipwd:</u> Password
- <u>text:</u> Message Content
- <u>to:</u> Recipient of message

Optional Variables:

<u>country:</u> 2 letter ISO code of country of recipient (eg: if you send a message to 07712345678, and specify the country UK, it will change the prefix to 4477. Your account has a default country code to which a 0 would be prefixed with, so setting the country prefix will override your accounts default setting)

<u>from:</u> Origin number (e.g., this will override your account's default origin number on a per message basis, please note that this field can be up to either **11 alphanumeric characters**, or **16 numeric characters**)

Code Examples:

PHP – Basic HTTP API Connect:

```php
<?php
//
// Required variables to send a message:
//
 $to="353871234567";
 $message=urlencode("hello");
 $username="sampleuser";
 $password="testuser";

//
// Optional variables
//
// country:   2 letter ISO code of country of recipient (eg: if you send a message to 07712345678, and specify the country UK,
//         it will change the prefix to 4477. your account has a default country code which a 0 would be prefixed with
//         Setting the country prefix will override your accounts default)
//         Example: $country="UK";
// from:     This will change your origin number dynamically per message.
//         This overrides your accounts default origin number
//         Example: $from="3535864000";

// Enable the line below to enable SSL. Dont forget to hash out the other baseurl.
// $baseurl="https://api.neonsolutions.ie";

 $baseurl="http://api.neonsolutions.ie";
 $url=$baseurl."/sms.php?user=$username&clipwd=$password&text=$message&to=$to";
 $return=file($url);
 $rtrnstr=explode(":",$return[0]);
 if ($rtrnstr[0] == 'OK') {
  $msgid=trim($rtrnstr[1]);
  echo $msgid;    // stuff to do after message gets sent.
 } else {
```

```
    $error=trim($rtrnstr[1]);
    echo $error;   // error handling routine goes here.
  }
?>
```

PHP -Advanced HTTP API Connect :

Supported Variables

- **user :** Username
- **clipwd :** Password
- **text :** Text message, this variable needs to be URL Encoded as per RFC 3986 http://tools.ietf.org/html/rfc3986
- **to :** Recipient, number should be in full international format, eg: 353871234567. Alternatively, a 0 prefix will be replaced with the user account's default country prefix, eg: 0871234567 will be replaced with 353871234567 if the account sending the message has Ireland set as it's default country prefix.
- **from (optional) :** This sets the origin number for the text message (*see notes above on length limitations*)
- **country (optional) :** This overrides the default country prefix for a single message, this should be represented as a 2 letter country code, eg: IE
- **date (optional) :** This variable will allow a message to be scheduled for any date in the future, the format we require for this is yyyymmddhhmmss, eg: 20080101120000 This variable must be 14 characters long or it will be ignored and the message will be submitted for immediate delivery.

Advanced PHP code using a GET request:

```
<?
//
// Required variables to send a message:
//
  $to="353871234567";
  $message=urlencode("hello");
  $username="sampleuser";
  $password="testuser";


//
// Optional variables
//
// country:  2 letter ISO code of country of recipient (eg: if you send a message to 07712345678, and specify the
//           country UK, it will change the prefix to 4477. your account has a default country code which a 0 would
//           be prefixed with. Setting the country prefix will override your accounts default)- Example:
//           $country="UK";
// from:     This will change your origin number dynamically per message.
//           This overrides your accounts default origin number
//           Example: $from="3535864000";
// date:     This variable will allow a message to be scheduled for any date in the future, the format we require for
//           this is yyyymmddhhmmss. This variable must be 14 characters long or it will be ignored and the
//           message delivered immediately - Example: $date="20070424180000";

// Enable the 2 lines below to enable SSL. Dont forget to hash out the other host & port.
// $host='ssl://api.neonsolutions.ie';
// $port=443;


  $host='api.neonsolutions.ie';
  $port=80;


  $sock = fsockopen("$host", $port, $errno, $errstr, 120);
  if(!$sock){
```

```php
      echo "Unable to get server status";
    }else{
      $out=sprintf("GET /sms.php?user=$username&clipwd=$password&text=$message&to=$to");
      $out.=" HTTP/1.1\r\n";
      $out .= "Host: api.neonsolutions.ie\r\n";
      $out .= "Connection: Close\r\n\r\n";
      fwrite($sock, $out);
      stream_set_blocking($sock, FALSE );
      stream_set_timeout($sock, 120);
      $info = stream_get_meta_data($sock);
      while (!feof($sock) && !$info['timed_out']) {
        $file .= fgets($sock, 4096);
        $info = stream_get_meta_data($sock);
      }
      $arr=explode("\n",$file);
      for ($i=0;$i < count($arr);$i++) {
        if (preg_match("/($errresp)|($okresp)/",$arr[$i])) {
          $return=$arr[$i];
        }
      }
      fclose($sock);
    }
    $rtrnstr=explode(":",$return);
    if ($rtrnstr[0] == 'OK') {
      $msgid=trim($rtrnstr[1]);
      // stuff to do after message gets sent.
    } else {
      // error handling routine goes here.
    }
?>
```

Advanced PHP code using a POST request:

```php
<?
//
// Required variables to send a message:
//
  $to="353871234567";
  $message=urlencode("hello");
  $username="sampleuser";
  $password="testuser";


//
// Optional variables
//
// country:  2 letter ISO code of country of recipient (eg: if you send a message to 07712345678, and specify the
//                country UK, it will change the prefix to 4477. your account has a default country code which a 0 would
//                be prefixed with Setting the country prefix will override your accounts default) - Example:
//                $country="UK";
// from:      This will change your origin number dynamically per message.
//                This overrides your accounts default origin number
//                Example: $from="3535864000";
// date:      This variable will allow a message to be scheduled for any date in the future, the format we require for
//                this is yyyymmddhhmmss. This variable must be 14 characters long or it will be ignored and the
//                message delivered immediately- Example: $date="20070424180000";


// Enable the 2 lines below to enable SSL. Dont forget to hash out the other host & port.
// $host='ssl://api.neonsolutions.ie';
// $port=443;
```

```php
  $host='api.neonsolutions.ie';
  $port=80;
  $sock = fsockopen("$host", $port, $errno, $errstr, 120);

  if(!$sock){
   echo "Unable to get server status";
  }else{
   $po="user=$username&clipwd=$password&text=$message&to[]=$to&to[]=$to2&to[]=$to3";
// in this example we can send the same message to 3 different numbers, the limit of this is over
// 4096, repeat the to[] variable for each recipient.
   $out=sprintf("POST /sms.php");
   $out.=" HTTP/1.1\n";
   $out .= "Content-Type: application/x-www-form-urlencoded\r\n";
   $out .= "Host: $host\r\n";
   $len=strlen($po);
   $out .= "Content-Length: $len\r\n";
   $out .= "Connection: Close\r\n";
   $out .= "Cache-Control: no-cache\r\n\r\n";
   $out .= $po;
   fwrite($sock, $out);
   stream_set_blocking($sock, FALSE );
   stream_set_timeout($sock, 120);
   $info = stream_get_meta_data($sock);
   while (!feof($sock) && !$info['timed_out']) {
     $file .= fgets($sock, 4096);
     $info = stream_get_meta_data($sock);
   }
   $arr=explode("\n",$file);
   for ($i=0;$i < count($arr);$i++) {
     if (preg_match("/($errresp)|($okresp)/",$arr[$i])) {
       $return=$arr[$i];
     }
   }
   fclose($sock);
  }
  $rtrnstr=explode(":",$return);
  if ($rtrnstr[0] == 'OK') {
    $msgid=trim($rtrnstr[1]);
// stuff to do after message gets sent.
  } else {
// error handling routine goes here.
  }
?>
```

Notes:
This method is more efficient as it works directly with the TCP/IP stack, and creates all the packets as needed from the start, this method also has the ability to specify your own timeouts per connection.

Specifying Connection: Close in the HTTP header allows the server to shutdown the connection as soon as it has process the message & returned the response.

C#* -Advanced HTTP API Connect :

```csharp
using System;
using System.Collections.Generic;
using System.Text;
using System.Text.RegularExpressions;
using System.Net.Sockets;

namespace SMS
{
   public class SMSSender
   {


      public void  SendSMS(string[] listOfReceipients, string message)
      {
         if (listOfReceipients.Length <= 0)
         {
            throw new Exception("The receipient list is empty");
         }
         int maxMsgLen = 459;
         if (message.Length > maxMsgLen)
            throw new Exception("Message length is "+message.Length+", which is greater than the allowed "+maxMsgLen);

         string userName = Settings1.Default.UserName;
         string passWord = Settings1.Default.Password;
         string originator = Settings1.Default.Originator;
         string listOfReplaceableChars = Settings1.Default.ReplaceableChars;
         StringBuilder receipientList = new StringBuilder();
         foreach (string  recpt in listOfReceipients)
         {

            receipientList.Append("&to[]="+Regex.Replace(recpt,listOfReplaceableChars,string.Empty));
         }

         string value =
"user="+userName+"&clipwd="+passWord+"&from="+originator+"&text="+message+receipientList.ToString();
         StringBuilder sb = new StringBuilder();
         sb.Append("POST /sms.php");
         sb.Append(" HTTP/1.1\n");
         sb.Append("Content-Type: application/x-www-form-urlencoded\r\n");
         sb.Append("Host: api.neonsolutions.ie\r\n");
         sb.Append("Content-Length: " + value.Length + "\r\n");
         sb.Append("Connection: Close\r\n");
         sb.Append("Cache-Control: no-cache\r\n\r\n");
         sb.Append(value);
         Connect("api.neonsolutions.ie", sb.ToString());


      }

      private void Connect(String server, String message)
      {
          NetworkStream stream ;
          TcpClient client;
          try
```

```csharp
        {

            Int32 port = Settings1.Default.SMSServerPort;
             client = new TcpClient(server, port);

            // Translate the passed message into ASCII and store it as a Byte array.
            Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

             stream = client.GetStream();
            stream.Write(data, 0, data.Length);

            data = new Byte[256];

            String responseData = String.Empty;


            Int32 bytes = stream.Read(data, 0, data.Length);
            responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
            // Close everything.
            stream.Close();
            client.Close();
            if (!responseData.Contains("200"))
            {
               throw new Exception("There was a problem sending the SMS messages\r\n"+responseData);
            }

        }

        catch (Exception e)
        {
         throw e;
        }
        finally
        {

        }
    }
    }
}
```

VBA - Basic HTTP API for integration with MS Office Applications

```vba
'URL Encoding Function – required for URL encoding of the sumbmitted message
'---------------------------------
Public Function URLEncode( _
    StringVal As String, _
    Optional SpaceAsPlus As Boolean = False _
) As String

    Dim StringLen As Long: StringLen = Len(StringVal)

    If StringLen > 0 Then
        ReDim Result(StringLen) As String
        Dim i As Long, CharCode As Integer
        Dim Char As String, Space As String

        If SpaceAsPlus Then Space = "+" Else Space = "%20"

        For i = 1 To StringLen
            Char = Mid$(StringVal, i, 1)
            CharCode = Asc(Char)
            Select Case CharCode
                Case 97 To 122, 65 To 90, 48 To 57
                    Result(i) = Char
                Case 32
                    Result(i) = Space
                Case 0 To 15
                    Result(i) = "%0" & Hex(CharCode)
                Case Else
                    Result(i) = "%" & Hex(CharCode)
            End Select
        Next i
        URLEncode = Join(Result, "")
    End If
End Function


'Sending a Text
'----------------

Sub httpSMS()
    'Declaring variables required for authentication and submission
    Dim message As String
    Dim user As String
    Dim clipwd As String
    Dim mobile As String
    Dim baseurl As String
    Dim getstring As String

    'Assigning the variables required for authentication and submission
    message = "Test message, remember string must be URL encoded!"
    user = "username"
    clipwd = "password"
    mobile = "353871234567"
```

```
    'IMPORTANT - URL enxcoding of text to be sent - without this step the text will not be sent or a blank string will
be submitted
    urlmessage = URLEncode(message, True)

    baseurl = "http://api.neonsolutions.ie/sms.php?"
    getstring = baseurl & "user=" & user & "&clipwd=" & clipwd & "&text=" & urlmessage & "&to=" & mobile

    'The GET string being submitted to NeonSMS
    MsgBox (getstring)

    'Declaring the object and setting up the GET request
    Dim NeonGetRequest As Object
    Set NeonGetRequest = CreateObject("WinHttp.WinHttpRequest.5.1")
    NeonGetRequest.Open "GET", _
    getstring

    ' Send Request.
    NeonGetRequest.Send

    'And we get this response
    MsgBox NeonGetRequest.ResponseText
End Sub
```

Support:

If you require any support on any aspects of integration with the Neon Solutions SMS Gateway, please contact our developers in the support department.

- **Email:**   support@neonsolutions.ie
- **Telephone:**   +353 21 2430700