SMS Gateway Interface

Format

Client should perform Http Post request to this URL.
The encoding should be UTF-8.

Using Dot Net Programming:

- Message text must be escaped with System.Security.SecurityElement.Escape.
- XML must be encoded by HttpUtility.UrlEncode(xml, .Text.Encoding.UTF8)

http://api.smsim.co.il/SendMessageXml.ashx?InforuXML={xml}

Simple Xml example

```
<Inforu>
  <User>
    <Username>MyUsername</Username>
    <Password>MyPassword</Password>
  </User>
  <Content Type="sms">
    <Message>This is a test SMS Message</Message>
  </Content>
  <Recipients>
    <PhoneNumber>0501111111;0502222222</PhoneNumber>
  </Recipients>
  <Settings>
    <SenderNumber>0501111111</SenderNumber>
  </Settings>
</Inforu>
```

Explanation of XML structure:

- **UserName** - Username of the account that was supplied by InforUMobile
- **Password** - Password of the account.
- **Message** – SMS message that need to be sent.
- **PhoneNumber** – the recipients phone list. Can be multiple recipients devided by ";"
- **SenderNumber** – The number that will be displayed on the recipient phone as the sender.

Advance Xml example

```
<Inforu>
  <User>
    <Username>MyUserName</Username>
    <Password>MyPassword</Password>
  </User>
  <Content Type="sms">
    <Message> This is a test SMS Message </Message>
    <MessagePelephone> This is a test SMS Message for Pelephone</MessagePelephone>
    <MessageCellcom> This is a test SMS Message for Cellcom </MessageCellcom>
    <MessageOrange> This is a test SMS Message for Orange </MessageOrange>
    <MessageMirs> This is a test SMS Message for Mirs </MessageMirs>
  </Content>
  <Recipients>
    <PhoneNumber>0501111111;0502222222</PhoneNumber>
    <GroupNumber>5</GroupNumber>
  </Recipients>
  <Settings>
    <SenderName>Inforu</SenderName>
    <SenderNumber>0501111111</SenderNumber>
    <CustomerMessageID>112233</CustomerMessageID>
    <CustomerParameter>AffId4</CustomerParameter>
    <MessageInterval>0</MessageInterval>
    <TimeToSend>12/3/2009 12:23</TimeToSend>
    <DeliveryNotificationUrl>http://mysite.co.il/Notif.aspx</DeliveryNotificationUrl>
    <MaxSegments>0</MaxSegments>
  </Settings>
</Inforu>
```

Explanation of XML structure (All the following parameters are **optional**):

- **SenderName** – The "SenderName" field is restricted to a maximum of 11 latin characters, and supported only in CELLCOM and ORANGE networks. In PELEPHONE and MIRS networks the "SenderName" will be replaced by "SenderNumber". Using the "SenderName" field overwrites the "SenderNumber" and not allowing the recipient to use the "reply" option on his phone. If omitted SenderNumber" will be used.
- **MessagePelephone –** Use this parameter to send a message to Pelephone recipients. If omitted the Message parameter will be use.
- **MessageCellcom –** Use this parameter to send a message to Cellcom recipients. If omitted the Message parameter will be use.
- **MessageOrange –** Use this parameter to send a message to Orange recipients. If omitted the Message parameter will be use.
- **MessageMirs -** Use this parameter to send a message to Mirs recipients. If omitted the Message parameter will be use.
- **GroupNumber -** Use this parameter in order to send message to group according to the group defined in the web site. To send to multiple group please use ";".
- **CustomerMessageID -** Message ID on the client application. When confirmation on delivery is sent back to the client, the message ID is sent also for synchronization.
- **CustomerParameter** - Parameter set by the client that can be seen later in reports. Can be used for example to mark each message with affiliate Id.
- **MessageInterval -** Sending messages with number of seconds interval between them The value 0 means not interval.
- **TimeToSend** – the requested date and time on which the messages will be sent. Please use the following format: dd/m/yyyy hh:mm

- **DeliveryNotificationUrl** - needed in order to send confirmation on delivery to the client. The system will perform Http post to this URL with notification information.
- **MaxSegments** – when sending long messages, this parameter allow the client to set the maximum number of segments per message. Value 0 mean- unlimited segments.

Response XML

Explanation of Response XML structure:

```
<Result>
    <Status></Status >
    <Description></ Description>
    <NumberOfRecipients></ NumberOfRecipients>
</Result>
```

1. **Status:**
    - OK=1,
    - Failed=-1,
    - BadUserNameOrPassword=-2,
    - UserNameNotExists=-3,
    - PasswordNotExists=-4,
    - RecipientsDataNotExists=-6,
    - MessageTextNotExists=-9,
    - IllegalXML=-11,
    - UserQuotaExceeded=-13,
    - ProjectQuotaExceeded=-14,
    - CustomerQuotaExceeded=-15,
    - WrongDateTime=-16,
    - WrongRecipients=-18,
    - InvalidSenderNumber=-20,
    - InvalidSenderName=-21,
    - UserBlocked=-22,
    - UserAuthenticationError=-26,
    - NetworkTypeNotSupported=-28,
    - NotAllNetworkTypesSupported=-29,
2. **Description –** action verbal description.
3. **NumberOfRecipients -** phone numbers that the message has been sent.

Delivery Notification

The client can receive confirmation on delivery on each message sent. When sending the message the client should use the CustomerMessageID and the DeliveryNotificationUrl parameters. The system perform Http Post request to this URL about any message.

The parameters for the request are:

- PhoneNumber – the number of the recipient.
- Network – the network of the recipient
- Status – the status of the message.
  - (2) – Delivered.
  - (-2) – Not delivered.
  - (-4) – Blocked by InforuMobile.
- StatusDescription – If not delivered, contains the reason.
- CustomerMessageId – the client message Id.
- SegmentsNumber - The amount of segments in the message

Delivery Notification Example

URL: http://www.clientURL.co.il/demo.aspx

With parameters: PhoneNumber=0521111111&Network=052&Status=2&StatusDescription=Message delivered&CustomerMessageId=111&SegmentsNumber=2

Message character length

Operator	Hebrew	English	Remarks
Cellcom	70	160	Supports concatenation. 70/160 chars – 1 message 71/154 – 500 – each message contain 67 chars Hebrew (154 chars English) but all the messages appears as one message on the phone.
Orange	70	160	Supports concatenation. 70/160 chars – 1 message 71/154 – 500 – each message contain 67 chars Hebrew (154 chars English) but all the messages appears as one message on the phone.
Pelephone	70	160	Does not support concatenation. Message will be divided to 126 chars in each segment.
Mirs	70	140	Does not support concatenation. Message will be divided to 70 chars Hebrew (140 chars English) in each segment.

C# example

```
//set password, user name, message text, semder name and number
string userName = "UserName";
string password = "UserPassword";
string messageText = System.Security.SecurityElement.Escape("Message text");
string senderName = "MySender";
string senderNumber = "1111";



//set phone numbers
string phonesList = "0503333333;0503333334;0503333335;0503333336;0503333337";

//set additional parameters
string timeToSend = "02/12/2008 15:30";

// create XML
StringBuilder sbXml = new StringBuilder();
sbXml.Append("<Inforu>");
sbXml.Append("<User>");
sbXml.Append("<Username>" + userName + "</Username>");
sbXml.Append("<Password>" + password + "</Password>");
sbXml.Append("</User>");
sbXml.Append("<Content Type=\"sms\">");
sbXml.Append("<Message>" + messageText + "</Message>");
sbXml.Append("</Content>");
sbXml.Append("<Recipients>");
sbXml.Append("<PhoneNumber>" + phonesList + "</PhoneNumber>");
sbXml.Append("</Recipients>");
sbXml.Append("<Settings>");
sbXml.Append("<SenderName>" + senderName + "</SenderName>");
sbXml.Append("<SenderNumber>" + senderNumber + "</SenderNumber>");
sbXml.Append("<MessageInterval>" + messageInterval + "</MessageInterval>");
sbXml.Append("<TimeToSend>" + timeToSend + "</TimeToSend>");
sbXml.Append("</Settings>");
sbXml.Append("</Inforu >");
string strXML = HttpUtility.UrlEncode(sbXml.ToString(), System.Text.Encoding.UTF8);
string result = PostDataToURL("http://api.smsim.co.il/SendMessageXml.ashx", "InforuXML=" + strXML);


static string PostDataToURL(string szUrl, string szData)
{
  //Setup the web request
  string szResult = string.Empty;
  WebRequest Request = WebRequest.Create(szUrl);
  Request.Timeout = 30000;
  Request.Method = "POST";
  Request.ContentType = "application/x-www-form-urlencoded";

   //Set the POST data in a buffer
  byte[] PostBuffer;
  try
  {
    // replacing " " with "+" according to Http post RPC
    szData = szData.Replace(" ", "+");

    //Specify the length of the buffer
```

```
      PostBuffer = Encoding.UTF8.GetBytes(szData);
      Request.ContentLength = PostBuffer.Length;

      //Open up a request stream
      Stream RequestStream = Request.GetRequestStream();

      //Write the POST data
      RequestStream.Write(PostBuffer, 0, PostBuffer.Length);

       //Close the stream
      RequestStream.Close();
     //Create the Response object
     WebResponse Response;
     Response = Request.GetResponse();

     //Create the reader for the response
    StreamReader sr = new StreamReader(Response.GetResponseStream(),     Encoding.UTF8);

     //Read the response
     szResult = sr.ReadToEnd();

      //Close the reader, and response
     sr.Close();
     Response.Close();

     return szResult;
  }
  catch (Exception e)
  {
        return szResult;
   }
}
```

PHP example

```
function send_sms($msg, $recepients)
{
$msg = str_replace('<', '%26lt;', $msg); // "Cleans" the message from unsafe notes
$msg = str_replace('>', '%26gt;', $msg); // "Cleans" the message from unsafe notes
$msg = str_replace('\"', '%26quot;', $msg); // "Cleans" the message from unsafe notes
$msg = str_replace("\'", '%26apos;', $msg); // "Cleans" the message from unsafe notes
$msg = str_replace("&", '%26amp;', $msg); // "Cleans" the message from unsafe notes
$message_text = $msg;
$sms_host = " api.smsim.co.il"; // Application server's URL;
$sms_port = 80; // Application server's PORT;
$sms_path = "/SendMessageXml.ashx"; // Application server's PATH;

// EDIT THE FOLLOWING LINES

$sms_user = "USERNAME"; // User Name (Provided by SMSIM) ;
$sms_password = "PASSWORD"; // Password (Provided by SMSIM) ;
$sms_sender_name = "MyCompany"; // The SMS sender's will appear only on Cellcom & Partner's phones
(Optional, only English characters, 11 max. );
$sms_sender_num = "1315"; // The number that this SMS will be sent from;
$message_text = str_replace (" ", "%20", $message_text); // Encodes spaces into "%20" in the URL;
$query = 'InforuXML=' . urlencode('<Inforu><User><Username>'.
$sms_user.'</Username><Password>'.
$sms_password.'</Password></User><Content Type="sms"><Message>').
$message_text.urlencode('</Message></Content><Recipients><PhoneNumber>'.
$recepients.'</PhoneNumber></Recipients><Settings>><SenderName>'.
$sms_sender_name.'</SenderName><SenderNumber>'.
$sms_sender_num.'</SenderNumber><CustomerParameter>'.
$customer_parameter.'</CustomerParameter></Settings></Inforu>');


$fp = fsockopen("$sms_host", $sms_port, $errno, $errstr, 30); // Opens a socket to the Application server

if (!$fp)
        { // Verifies that the socket has been opened and sending the message;
        echo "$errstr ($errno)<br />\n";
        } else
        {
        $out = "GET $sms_path?$query HTTP/1.1\r\n";
        $out .= "Host: $sms_host\r\n";
        $out .= "Connection: Close\r\n\r\n";
        fwrite($fp, $out);
        while (!feof($fp))
                {
                echo fgets($fp, 128); // Echos the respond from application server (you may replace this line
with an "Message has been sent" message);
                }
        fclose($fp);
        }
}
```

VB example

Important: need to add reference to System.Web.dll

```
'set password, user name, message text, semder name and number
Dim userName As String = "UserName"
Dim password As String = "UserPassword"
Dim messageText As String = System.Security.SecurityElement.Escape("Message text")
Dim senderName As String = "MySender"
Dim senderNumber As String = "1111"
Dim messageInterval As Integer = 0

'set phone numbers
Dim phonesList As String = "0503333333;0503333334;0503333335;0503333336;0503333337"

'set additional parameters
Dim timeToSend As String = "02/12/2008 15:30"

' create XML
Dim sbXml As New Text.StringBuilder()
sbXml.Append("<Inforu>")
sbXml.Append("<User>")
sbXml.Append("<Username>" & userName & "</Username>")
sbXml.Append("<Password>" & password & "</Password>")
sbXml.Append("</User>")
sbXml.Append("<Content Type=""sms"">")
sbXml.Append("<Message>" & messageText & "</Message>")
sbXml.Append("</Content>")
sbXml.Append("<Recipients>")
sbXml.Append("<PhoneNumber>" & phonesList & "</PhoneNumber>")
sbXml.Append("</Recipients>")
sbXml.Append("<Settings>")
sbXml.Append("<SenderName>" & senderName & "</SenderName>")
sbXml.Append("<SenderNumber>" & senderNumber & "</SenderNumber>")
sbXml.Append("<MessageInterval>" & messageInterval & "</MessageInterval>")
sbXml.Append("<TimeToSend>" & timeToSend & "</TimeToSend>")
sbXml.Append("</Settings>")
sbXml.Append("</Inforu >")

Dim strXML As String = System.Web.HttpUtility.UrlEncode(sbXml.ToString(), System.Text.Encoding.UTF8)
Dim result As String = PostDataToURL("http://api.smsim.co.il/SendMessageXml.ashx", "InforuXML=" &
strXML)

Private Function PostDataToURL(ByVal szUrl As String, ByVal szData As String) As String
        'Setup the web request
        Dim szResult As String = String.Empty
        Dim Request As Net.HttpWebRequest
        Request = CType(System.Net.WebRequest.Create(szUrl), System.Net.HttpWebRequest)
        Request.Timeout = 30000
        Request.Method = "POST"
        Request.ContentType = "application/x-www-form-urlencoded"

        'Set the POST data in a buffer
        Dim PostBuffer As Byte()
        Try
                ' replacing " " with "+" according to Http post RPC
                szData = szData.Replace(" ", "+")
                'Specify the length of the buffer
                PostBuffer = System.Text.Encoding.UTF8.GetBytes(szData)
                Request.ContentLength = PostBuffer.Length

                'Open up a request stream
                Dim RequestStream As IO.Stream = Request.GetRequestStream()

                'Write the POST data
                RequestStream.Write(PostBuffer, 0, PostBuffer.Length)

                'Close the stream
                RequestStream.Close()
                'Create the Response object
                Dim Response As Net.HttpWebResponse
                Response = CType(Request.GetResponse(), System.Net.HttpWebResponse)

                'Create the reader for the response
                Dim sr As New IO.StreamReader(Response.GetResponseStream(),
System.Text.Encoding.UTF8)

                'Read the response
                szResult = sr.ReadToEnd()

                'Close the reader, and response
                sr.Close()
                Response.Close()

                Return (szResult)
        Catch e As Exception
                Return (szResult)
        End Try
End Function
```