Version 3.0

SMS Gateway Interface

Send Sms using http post Interface

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

http://smsapi.hadavar.co.il/SendMessageXml.ashx?InforuXML={xml}

Secure interface

For a secure interface use:

https://smsapi.hadavar.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>
    <Sender>0501111111</Sender>
  </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 ";"
- **Sender** – The id that will be displayed on the recipient phone as the sender.

Advance Xml example

```xml
<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>
    <GroupNumber>5</GroupNumber>
  </Recipients>
  <Settings>
    <Sender>Inforu</Sender>
    <CustomerMessageID>112233</CustomerMessageID>
    <CustomerParameter>AffId4</CustomerParameter>
    <MessageInterval>0</MessageInterval>
    <TimeToSend>12/05/2013 12:23</TimeToSend>
    <DelayInSeconds>60</DelayInSeconds>
    <DeliveryNotificationUrl>http://mysite.co.il/Notif.aspx</DeliveryNotificationUrl>
    <MaxSegments>0</MaxSegments>
  </Settings>
</Inforu>
```

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

- **Sender** – The "Sender" field is restricted to a maximum of 11 latin characters (Consecutive characters without spaces). or 14 numbers only (no characters) You can also write a * at the beginning of the identifier.
- **Message** – SMS message that need to be sent.
- **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.
- **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.
- **TimeToSend** – Requested date and time on which the messages will be sent
  Please use the following format: dd/m/yyyy hh:mm. If left blank, the maesage will be sent immediately
- **DelayInSeconds** – Number of seconds delay before sending the message.

Response XML

Explanation of Response XML structure:

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

1. **Status:**
   - OK=1,
   - Failed=-1,
   - Bad user name or password=-2,
   - UserNameNotExists=-3,
   - PasswordNotExists=-4,
   - RecipientsDataNotExists=-6,
   - MessageTextNotExists=-9,
   - IllegalXML=-11,
   - UserQuotaExceeded=-13,
   - ProjectQuotaExceeded=-14,
   - CustomerQuotaExceeded=-15,
   - WrongDateTime=-16,
   - No valid recipients=-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.

Http Post code sample

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 sender = "MySender";



//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("<Sender>" + sender+ "</Sender>");
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://smsapi.hadavar.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;
     }
  }
```

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)

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
$msg = str_replace("\r\n", '%0D%0A', $msg);   // "Cleans" the message from enter
$message_text = $msg;
$sms_host = "smsapi.hadavar.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 Inforu) ;
$sms_password = "PASSWORD"; // Password (Provided by Inforu) ;
$sms_sender = "MyCompany"; // The SMS sender's (Optional, only English characters, 11 max. );
$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><Sender>'.
$sms_sender.'</Sender><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 sender As String = "MySender"
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("<Sender>" & sender & "</Sender>")
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://smsapi.hadavar.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
```

Send SMS using Web Service Interface

Description

Web service interface is mainly used for .Net applications although Java developers can use it too. Microsoft .Net framework has a built in supports for calling web services so for anyone who uses .Net it is highly recommended to use this interface. This interface allows the user to control and customize every aspect of InforuMobile SMS engine.

Format

Microsoft .Net users should add reference to this URL:
http://smsapi.hadavar.co.il/SendMessage.asmx

Java / Delphi / Magic users should use this URL:
http://smsapi.hadavar.co.il/SendMessage.asmx?wsdl

The web interface implements 2 functions:

- SendSms – sending sms with minimum parameters
- SendSmsDetailed – sending sms with additional parameters

Function parameters explanation

There are 2 functions for sending SMS messages.

Here is an explanation of the parameters from the function **SendSmsDetailed** :

- **UserName** - Username of the account that was supplied by InforUMobile
- **Password** - Password of the account.
- **Message** – SMS message that need to be sent.
- **MessagePelephone –** Use this parameter to send a message to Pelephone recipients. If omitted the Message parameter will be used.
- **MessageCellcom –** Use this parameter to send a message to Cellcom recipients. If omitted the Message parameter will be used.
- **MessageOrange –** Use this parameter to send a message to Orange recipients. If omitted the Message parameter will be used.
- **MessageMirs -** Use this parameter to send a message to Mirs recipients. If omitted the Message parameter will be used.
- **Recipients** – Recipients phone list. Multiple recipients divided by ";"
- **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.
- **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.
- **MessageInterval -** Sending messages with number of seconds interval between them. The value 0 means no interval.

- **TimeToSend** – the requested date and time on which the messages will be sent. Please use the following format: dd/m/yyyy hh:mm
- **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.
- **SenderNumber** – The number that will be displayed on the recipient phone as the sender.
- **MaxSegments** – when sending long messages, this parameter allow the client to set the maximum number of segments per message. Value 0 mean- unlimited segments.

Successful response example

```
<Result>
<Status>1</Status>
<Description>Message accepted successfully</Description>
<NumberOfRecipients>1</NumberOfRecipients>
</Result>
```

Unsuccessful response example

```
<Result>
<Status>-2</Status>
<Description>Error: Bad user name or password</Description>
<NumberOfRecipients>0</NumberOfRecipients>
</Result>
```

Response status explanations

1 - OK

-1 - Failed

-2 - Bad User Name Or Password

-3 - User Name Not Exists

-4 - Password Not Exists

-6 - No Recipients

-9 - Message Text Not Exists

-13 - User Quota Exceeded

-14 - Project Quota Exceeded

-15 - Customer Quota Exceeded

-16 - Wrong Date Time

-17 - Wrong Number Parameter

-18 - Wrong Recipients

-20 - Invalid Sender Number

-21 - Invalid Sender Name

-22 - User Blocked

Message character length

Upated information can be found here

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

Receiving SMS (optional)

When a user sends an SMS from his cell phone, the system will perform Http Post Request to the client predefined URL. The client receives the XML in the “IncomingXML” parameter, encoded in UTF-8.

For example:

Request: - To Your Url

IncomingXML=

```
<IncomingData>
        <PhoneNumber>0501111111</PhoneNumber>
        <Keyword>Hello</Keyword>
        <Message>Hello Word</Message>
        <Network>052</Network>
        <ShortCode>5454</ShortCode>
        <CustomerID>12</CustomerID>
        <ProjectID>123</ProjectID>
        <ApplicationID>12321</ApplicationID>
</IncomingData>
```

Example

```
http://ClientWebApp.co.il/GetSMS.aspx?IncomingXML=
<IncomingData>
<PhoneNumber>0509999111</PhoneNumber>
<Keyword>Yes</Keyword>
<Message> Yes I am interested </Message>
<Network>054</Network>
<ShortCode>039444666</ShortCode>
<CustomerID>17</CustomerID>
<ProjectID>456</ProjectID>
<ApplicationID>33321</ApplicationID>
</IncomingData>
```

Response:

O.K.

Tools

Delete Future Message

You can delete future sent messages programmatically using a WebService.

URL WebService:

http://smsapi.hadavar.co.il/ClientServices.asmx

Function name : DeleteFutureMessage

Parameters:

UserName – Username of the user

Password – User's password

phoneNumber – Number to which the message was sent

customerMessageId– ID sent in sending the message

customerParam - Parameter sent in sending the message

Example:

UserName – david

Password – 123456

phoneNumber - 05021111111

customerMessageId – a1b2c

customerParam – poi123

response:

```
<ClientNotification>
<Status> </Status>
<BatchSize></BatchSize>
</ClientNotification>
```

Explanation of those parameters:

OK -Success
Failed -Failed
BatchSize – Many numbers are deleted

Remaining SMS Quota

A function that returns the amount of SMS remaining in the user account.

URL WebService:

http://smsapi.hadavar.co.il/WebTools.asmx

Function name : GetUserQuota

URL GET or POST:

http://smsapi.hadavar.co.il/WebTools/GetUserQuota.ashx

Parameters:

UserName – Username of the user

Password – User's password

An example of a request in Http Get:

http://smsapi.hadavar.co.il/WebTools/GetUserQuota.ashx?userName=XXX&password=XXX

response:

```
<Result>
        <Status></Status>
        <Data></Data>
</Result>
```

Phone Number block check

The function checks in a certain phone number is blocked for receiving SMS.

URL WebService:

http://smsapi.hadavar.co.il/WebTools.asmx

Function name : CheckIfNumberBlocked

URL GET or POST:

http://smsapi.hadavar.co.il/WebTools/CheckIfNumberBlocked.ashx

Parameters:

UserName – Username of the user

Password – User's password

phoneNumber – Phone number (string)

prepaidAmount – Maximum charges is prepaid subscribers (integer)

An example of a request in Http Get:

http://smsapi.hadavar.co.il/WebTools/CheckIfNumberBlocked.ashx?userName=XXX&password=XXX&phoneNumber=0529999999&prepaidAmount=10

response:

```
<Result>
<Status></Status>
<Data></Data>
</Result>
```

Explanation of those parameters:

"-26" - Error User ID, Data empty
"-44" - No permission to access, Data empty
"-49" - Phone number is blocked, Data empty
"1"- Phone number is open, empty Data