Email Marketing & SMS Solution

Quick Start

SLNG Basic API Instruction

Version – 8.2

Contents

1. Introduction

This document is a quick start user guide for SLNG API application for sending SMS.

This document does not cover the complete API functionality, for further information please contact SLNG support team.

To start to work with API, please contact SLNG support team to get the API username and password.

In this document you can find detail instruction of the following topics:

Http post Interface:

1. Sending SMS messages via HTTP post
2. Getting Delivery Response (DLR) for SMS messages via HTTP post
3. Getting HTTP response for incoming SMS message
4. Get the user balance via HTTP post

Web service Interface:

1. Sending SMS messages
2. Get the user balance via HTTP post

2. Sending SMS using HTTP Post with JSON

HTTP post request should be sent to the following URL with UTF-8 encoding in the HTTP post request body.

2.3 Sending

Request/Command should be sent to the following URL with UTF-8 encoding.
http://slng5.com/Api/SendSmsJsonBody.ashx
Or (if SSL is needed)
https://slng5.com/Api/SendSmsJsonBody.ashx

Below section described the basic and the most common JSON format for Sending SMS message, if further capability are needed please contact SLNG support team.

```
{

        "Username": "xxxx",
         "Password": "xxx",
         "MsgName": "Test SLNG API",
         "MsgBody": "THIS IS A TEST MESSAGE",
         "FromMobile": "xxxxx",
         "DeliveryAckUrl": null,
         "MsgScheduleTime": null,
         "Mobiles": [
           {
                "Mobile": "xxxx",
                 "AckID": null,
                 "FirstName" : null,
                  "LastName" : null,
                  "Street" : null,
                   "City" : null,
                   "Country" : null,
                   "PostalCode" : null,
                   "Fax" : null
           },
           {
                 "Mobile": "xxxx",
                 "AckID": null,

                 "FirstName" : null,
                  "LastName" : null,
                  "Street" : null,
                   "City" : null,
                   "Country" : null,
                   "PostalCode" : null,
                   "Fax" : null
           }
          ]

     }
```

Fields descriptions:

- Username – your API username for SLNG system
- Password – your API password for SLNG system
- MsgName – Defines the message name that will present in SLNG system.
- MsgBody – your SMS message body will be here
- FromMobile – Defines the SMS sender information, this field can be a phone number or up to 12 characters in English.
- DeliveryAckURL – Optional field to define the URL for DLR (delivery response), See DLR section for further information
- MsgScheduleTime – Provide the option to schedule the message for future sending. Time format: DD/MM/YYYY HOUR:MINUTES. For example: The following will schedule the message to be sent on 26th of Aug at 10:35: "26/08/2018 10:35
- Mobile s– destination mobile object, this object define the mobile number as well as the customer AckID
  - Mobile – the actual destination mobile number
  - AckID - Optional field defines the customer AckID that will be sent back with DLR. See DLR for further information
  - FirstName – Optional field defines the FirstName of the contact and can be used for personalized message. See Sending personalized message for further information
  - LastName – Optional field defines the FirstName of the contact and can be used for personalized message. See Sending personalized message for further information
  - Street – Optional field defines the FirstName of the contact and can be used for personalized message. See Sending personalized message for further information
  - City– Optional field defines the FirstName of the contact and can be used for personalized message. See Sending personalized message for further information
  - Country – Optional field defines the FirstName of the contact and can be used for personalized message. See Sending personalized message for further information
  - PostalCode – Optional field defines the FirstName of the contact and can be used for personalized message. See Sending personalized message for further information
  - Fax – Optional field defines the FirstName of the contact and can be used for personalized message. See Sending personalized message for further information

2.4 JSON Response for Sending SMS Command

This section described the JSON object response for sending SMS message command via HTTP POST with JSON format.
Note that this response only confirm that command received successfully by SLNG. To get the status from the cellular companies please check the Delivery Response (DLR) section.

Return JSON Object:

```
{
        "Status":true/false,
        "Description":"Command received successfully",
        "GeneralGUID":"xxxxxx"
}
```

Field descriptions:

- Status –
  - True – Command received successfully
  - False - Error , see details below.
- Description – provide the error description.
- GeneralGUID – return the MsgInfoGUID of the current command, needed for CancelSendingCommand. See more details here.

Error message can be for examples one of the following:

- Username and/or Password are not valid
- Your account is closed, please contact SLNG support team
- Username has not been defined,please check your JSON string
- Password has not been defined,please check your JSON string
- Message name has not been defined,please check your JSON string
- Message body has not been defined,please check your JSON string
- FromMobile has not been defined,please check your JSON string
- Destination mobile numbers have not been defined,please check your JSON string

2.5 Sending Personalized SMS message

SLNG allow you to send a personalized SMS message via the API interface.

In the table below you can find the maximum length of each message and the format of the personalized field in the message body.

Please also be aware that you can define your own fields if needed, for further information please contact SLNG support team.

	Field Name (API)	Max length	Field in Message
1	FirstName	200	##FIRST_NAME## OR ##שם_פרטי##

2	LastName	200	##LAST_NAME## OR ##שם_משפחה##
3	Street	200	##STREET## OR ##רחוב ##
4	City	150	##CITY## OR ##עיר ##
5	Country	150	##COUNTRY## OR ##מדינה ##
6	PostalCode	20	##POSTAL_CODE## OR ##מיקוד##
7	Fax	50	##FAX## OR ##פקס ##

2.6 Example - Sending SMS with JSON object

2.6.1 PHP Example

```
<?php
function TestApiXml ()
{
     $url = "http://slng5.com/Api/SendSmsJsonBody.ashx";
     $msg_body = addslashes("THIS IS A TEST MESSAGE FROM SLNG");
     $json ='{
          "Username": "xxxxx",
          "Password": "xxxxxxx",
          "MsgName": $msg_body,
          "MsgBody": "THIS IS A TEST MESSAGE",
          "FromMobile": "0676151559",
          "DeliveryAckUrl": null,
          "Mobiles": [ { "Mobile": "xxxxxxxx"}]
          }';

     //-----------------------------------------------
     $CR = curl_init();
     curl_setopt($CR, CURLOPT_URL, $url);
     curl_setopt($CR, CURLOPT_POST, 1);
     curl_setopt($CR, CURLOPT_FAILONERROR, true);
     curl_setopt($CR, CURLOPT_POSTFIELDS, $json);
     curl_setopt($CR, CURLOPT_RETURNTRANSFER, 1);
     curl_setopt($CR, CURLOPT_HTTPHEADER,array("charset=utf-8"));
     //-----------------------------------------------
     $result = curl_exec($CR);
     $error = curl_error ($CR);
     //$response =  new SimpleXMLElement($result);
$response =  json_decode(urldecode($result));
}
TestApiJson();

?>
```

2.6.2 C# Example

Example below use Json.NET library to Serialize and De-serialize JSON objects.

Json.NET library can be downloaded from the following link:

http://www.newtonsoft.com/json

Further instruction about the installation of the package/library in Visual Studio environment can be found here:
https://docs.microsoft.com/en-us/nuget/tools/package-manager-console

```
using Newtonsoft.Json;// Json.NET library

       internal class SmsMobileJson
       {
            private string _Mobile;
            public string Mobile { get { return _Mobile; } set { _Mobile = value; } }

            private string _AckID;
            public string AckID { get { return _AckID; } set { _AckID = value; } }

            public SmsMobileJson()
            {
                this.AckID = null;
                this._Mobile = null;
            }
        }
        internal class SendSmdCmdJson
        {
            private string _Username;
            public string Username { get { return _Username; } set { _Username =
            value; } }

            private string _Password;
            public string Password { get { return _Password; } set { _Password =
            value; } }

            private string _MsgName;
            public string MsgName { get { return _MsgName; } set { _MsgName = value; }
            }

            private string _MsgBody;
            public string MsgBody { get { return _MsgBody; } set { _MsgBody = value; }
            }

            private string _FromMobile;
            public string FromMobile { get { return _FromMobile; } set { _FromMobile
             =value; } }

            private string _DeliveryAckUrl;
            public string DeliveryAckUrl { get { return _DeliveryAckUrl; } set {
            _DeliveryAckUrl = value; } }

            private string _MsgScheduleTime;
            public string MsgScheduleTime { get { return _MsgScheduleTime; } set {
            _MsgScheduleTime = value; } }

            private List<SmsMobileJson> _Mobiles;
            public List<SmsMobileJson> Mobiles { get { return _Mobiles; } set {
            _Mobiles = value; } }

            public SendSmdCmdJson()
            {
                this.MsgScheduleTime = null;
                this.DeliveryAckUrl = null;
                this.FromMobile = null;
                this.Mobiles = new List<SmsMobileJson>();
                this.MsgBody = null;
                this.MsgName = null;
                this.Password = null;
                this.Username = null;
            }
        }
        public class SmsResponseJson
        {
            private bool _Status;
            public bool Status { get { return _Status; } set { _Status = value; } }

            private string _Description;
            public string Description { get { return _Description; } set {
            _Description = value; } }

            public SmsResponseJson()
            {
                this.Status = false;
                this.Description = null;
            }
        }


       public static bool TestApiJson2()
       {
            string username = "xxxxx";
            string password = "xxxxx";
            string msg = System.Security.SecurityElement.Escape("Test SLNG API
##שם_פרטי## AND ##שם_משפחה## AND NOW INENGLISH: ##FIRST_NAME## AND ##LAST_NAME##");
            string sender = "077-4405268";
            string msgName = "Test SLNG API";

            List<string> destNo = new List<string>();
            destNo.Add("xxxxx");
            destNo.Add("xxxxx");

            SendSmdCmdJson obj = new SendSmdCmdJson();
            obj.Username = username;
            obj.Password = password;
            obj.MsgName = msgName;
            obj.MsgBody = msg;
            obj.FromMobile = sender;
            obj.DeliveryAckUrl = null;
            obj.HighPriority = false;
            foreach (string mobile in destNo)
            {
                SmsMobileJson obj1 = new SmsMobileJson();
                obj1.Mobile = mobile;
                obj1.FirstName = "Shlomi";
                obj1.LastName = "Haddad";
                obj.Mobiles.Add(obj1);
            }

            string json = JsonConvert.SerializeObject(obj, Formatting.Indented);

            string json_enc = System.Web.HttpUtility.UrlEncode(json.ToString(),
System.Text.Encoding.UTF8);
            SmsResponseJson obj2 = new SmsResponseJson();
            obj2 =
PostJsonDataToSLNG_Body("http://slng6.com/Api/SendSmsJsonBody.ashx", json_enc);
            if (obj2.Status)
                return true;
            else
                return false;

        }

        public static SmsResponseJson PostJsonDataToSLNG_Body(string url, string json)
        {

            //Setup the web request
            string result = string.Empty;
            WebRequest Request = WebRequest.Create(url);
            Request.Timeout = 30000;
            Request.Method = "POST";
            Request.ContentType = "application/x-www-form-urlencoded";
            //Set the POST data in a buffer
            byte[] xml_encoding;

            json = json.Replace(" ", "+");
            //Specify the length of the buffer
            xml_encoding = Encoding.UTF8.GetBytes(json);
            Request.ContentLength = xml_encoding.Length;
            //Open up a request stream
            Stream RequestStream = Request.GetRequestStream();
            //Write the POST data
            RequestStream.Write(xml_encoding, 0, xml_encoding.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
            result = sr.ReadToEnd();
            //Close the reader, and response
            sr.Close();
            Response.Close();
            string json_resp = System.Web.HttpUtility.UrlDecode(result);
            return JsonConvert.DeserializeObject<SmsResponseJson>(json_resp); ;

        }
```

3. DLR for SMS message

DLR (Delivery response) for SMS messages allow you to get HTTP post request to the DeliveryAckURL address (defines in the SMS command). Delivery response carries the status of the SMS message that received by the cellular companies.

The HTTP post parameter contain the following parameter.

- PHONE_NUMBER– the number of the recipient define in the XML mobile section.
- STATUS– the status of the message. See the table below for further info.
- REASON_MSG – If not delivered, contains the reason.
- CUSTOMER_MSG_ID – this field carry the AckID field , defines in the SMS XML command and it can be used to associate the return DLR with the client.
- MESSAGE_COUNT - The amount of segments in the message.
  For example, if your account is 70 characters account and your message is 101 character => MESSAGE_COUNT will be 2.

DLR Status

The table below described the DLR status supported by SLNG.

Status	Description
0	Message delivered
14	Subscriber is unavailable or message inbox full
747	Subscriber is abroad
4	Hard error, the number does not exist
16	Night Block feature – account cannot send messages in the predefined hours
15	Hard error, Kosher number blocked for SMS messages
2	Timeout- the number is currently not available
105	Operator error - number has not been blocked
555	Invalid number, block by SLNG
556	Unsubscribed number
557	Number with hard error record - block by SLNG
558	Mobile number is not defined (null or empty string)

2.7 DLR Example

Below example describe the HTTP POST request for the following SMS command:

- DeliveryAckURL = http://slng_example.com (defines in the SMS command)
- AckID = 5555 (optional field defines in the SMS command per client)
- Mobile = xxx (client phone number, defines in the SMS command)

Http POST:

http://slng_example.com? PHONE_NUMBER= xxx&STATUS=1&CUSTOMER_MSG_ID=555& REASON_MSG=deliverd&MESSAGE_COUNT=1

4. Incoming SMS message from Virtual Number

The system allows you to receive incoming SMS messages to the system and also to deliver the message to you over API via HTTP post interface.

To receive the incoming SMS messages via HTTP post request you need to provide us the URL.

Following parameter will be defined in the Http POST request:

- VIRTUAL_NUMBER - Your virtual number associate with this message
- CUSTOMER_NUMBER
- MSG – Receive message

2.7.1 Incoming SMS message – Example

Below example describe the HTTP POST request for incoming SMS message from virtual number.

- URL = hrrp://slng.example.com – define in the system for each virtual number

http://slng_example.com? VIRTUAL_NUMBER=0555555555&
CUSTOMER_NUMBER=xxx&MSG=example

In the above example we assumed that the incoming message received from mobile number 'xxx' and the virtual number is 0555555555

5. Get Message statistics using HTTP JSON Post Interface

HTTP post request should be sent to the following URL with UTF-8 encoding in the HTTP post request body.

http://slng5.com/Api/GetMsgStatistics.ashx

Or (if SSL is needed)
https://slng5.com/Api/GetMsgStatistics.ashx

4.2 Get message statistics JSON format

Below JSON format described the get message statistics command. As mentioned before the following JSON format should be sent in the request body.

Note that MsgInfoGUID field return in the response object of sending email command.

Request JSON:

```
{
  "Username": "xxxxxxxxx",
  "Password": "xxxxxxxxx",
  "MsgInfoGUID": "xxxxxxxxx"
}
```

Response JSON:

```
{

      "Pass": true,
      "Msg"   :"Command completed successfully",
       "TotalContact":"5",
       "TotalSent":"5",
       "TotalOpen":"5",
       "TotalError":"0",
       "TotalReturn":"0",
        "TotalUnsubscribe":"0",
        "TotalClick":"0",
        "TotalContactClick":"0"
}
```

4.3 Fields Description Request

Field Name	Description	Type	Required/Optional
Username	your API username for SLNG system	String	Required
Password	your API password for SLNG system	String	Required
MsgInfoGUID	MsgInfoGUID field return from the sending email command.	String	Required

4.4 Fields Description Response

Field Name	Description	Type
Pass	Return true or false value indicate the command status	Boolean
Msg	Described the message status	String
TotalContact	Total number of contacts that associated with this message	String
TotalSent	Total number of email that sent (usually it should be equal to the total number of contacts unless the sending message is still in progress).	String
TotalOpen	Total contacts that open the message	String
TotalError	Total error emails in the list (hard error)	String
TotalReturn	Total email that sent but return (soft error)	String
TotalUnsubscribe	Total unsubscribed contacts.	String
TotalClick	Total clicks on the message	String
TotalContactClick	Total contact that click on at list one link in the message	String

4.5 Example for getting message statistics using HTTP post

4.5.1 HTTP Post – PHP example

```
<?php
function SendEmailJson ()
{
  $url = " http://slng5.com/Api/GetMsgStatistics.ashx";
  $json ='{
  "Username": "xxxxxxxxx",
  "Password": "xxxxxxxxx",
  "MsgInfoGUID": "xxxxxxxxx"
}';
     //-----------------------------------------------
     $CR = curl_init();
     curl_setopt($CR, CURLOPT_URL, $url);
     curl_setopt($CR, CURLOPT_POST, 1);
     curl_setopt($CR, CURLOPT_FAILONERROR, true);
     curl_setopt($CR, CURLOPT_POSTFIELDS, $json);
     curl_setopt($CR, CURLOPT_RETURNTRANSFER, 1);
     curl_setopt($CR, CURLOPT_HTTPHEADER,array("charset=utf-8"));
     //-----------------------------------------------
     $result = curl_exec($CR);
     $error = curl_error ($CR);
     $response = new SimpleXMLElement($result);
}
TestApiJson();

?>
```

4.5.2 HTTP Post – C# example

Example below use Json.NET library to Serialize and De-serialize JSON objects.

Json.NET library can be downloaded from the following link:

http://www.newtonsoft.com/json

Further instruction about the installation of the package/library in Visual Studio environment can be found here:
https://docs.microsoft.com/en-us/nuget/tools/package-manager-console

```
using Newtonsoft.Json;// Json.NET library

   public static bool TestGetMsgStatisticsApiJson()
        {
            MsgStatisticsReqJson obj = new MsgStatisticsReqJson();
            MsgStatisticsResponseJson obj1 = new MsgStatisticsResponseJson();
            obj.Username = "xxxxxxxx";
            obj.Password = "xxxxxxxx";
            obj.MsgInfoGUID = "xxxxxxxxxxxxxxxxxx";
            string json = JsonConvert.SerializeObject(obj, Formatting.Indented);

            string json_enc = System.Web.HttpUtility.UrlEncode(json.ToString(), 
System.Text.Encoding.UTF8);
            string json_resp;
            json_resp = 
PostJsonBodyDataToSLNG("http://slng5.com/Api/GetMsgStatistics.ashx", json_enc);

            obj1 = 
JsonConvert.DeserializeObject<MsgStatisticsResponseJson>(json_resp);
            if (obj1.Pass)
                return true;
            else
                return false;

        }

  internal class MsgStatisticsReqJson
        {
            private string _Username;
            public string Username { get { return _Username; } set { _Username = 
value; } }

            private string _Password;
            public string Password { get { return _Password; } set { _Password = 
value; } }

            private string _MsgInfoGUID;
            public string MsgInfoGUID { get { return _MsgInfoGUID; } set { 
_MsgInfoGUID = value; } }

            public MsgStatisticsReqJson()
            {
                this._Username = null;
                this._Password = null;
                this.MsgInfoGUID = null;
            }

        }



  internal class MsgStatisticsResponseJson
        {
            private bool _Pass;
            public bool Pass { get { return _Pass; } set { _Pass = value; } }

            private string _Msg;
            public string Msg { get { return _Msg; } set { _Msg = value; } }

            private string _TotalContact;
            public string TotalContact { get { return _TotalContact; } set { 
_TotalContact = value; } }
```

```
            private string _TotalSent;
            public string TotalSent { get { return _TotalSent; } set { _TotalSent =
value; } }

            private string _TotalOpen;
            public string TotalOpen { get { return _TotalOpen; } set { _TotalOpen =
value; } }


            private string _TotalError;
            public string TotalError { get { return _TotalError; } set { _TotalError =
value; } }

            private string _TotalReturn;
            public string TotalReturn { get { return _TotalReturn; } set {
_TotalReturn = value; } }

            private string _TotalUnsubscribe;
            public string TotalUnsubscribe { get { return _TotalUnsubscribe; } set {
_TotalUnsubscribe = value; } }

            private string _TotalClick;
            public string TotalClick { get { return _TotalClick; } set { _TotalClick =
value; } }

            private string _TotalContactClick;
            public string TotalContactClick { get { return _TotalContactClick; } set {
_TotalContactClick = value; } }

            public MsgStatisticsResponseJson()
            {
                this._Msg = null;
                this._Pass = false;
                this._TotalContact = null;
                this.TotalClick = null;
                this._TotalContactClick = null;
                this.TotalError = null;
                this.TotalOpen = null;
                this.TotalReturn = null;
                this.TotalSent = null;
                this.TotalUnsubscribe = null;

            }

        }



      public static string PostJsonBodyDataToSLNG(string url, string json)
        {

            //Setup the web request
            string result = string.Empty;
            WebRequest Request = WebRequest.Create(url);
            Request.Timeout = 30000;
            Request.Method = "POST";
            Request.ContentType = "application/x-www-form-urlencoded";
            //Set the POST data in a buffer
            byte[] xml_encoding;

            json = json.Replace(" ", "+");
            //Specify the length of the buffer
            xml_encoding = Encoding.UTF8.GetBytes(json);
            Request.ContentLength = xml_encoding.Length;
            //Open up a request stream
            Stream RequestStream = Request.GetRequestStream();
            //Write the POST data
            RequestStream.Write(xml_encoding, 0, xml_encoding.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
            result = sr.ReadToEnd();
            //Close the reader, and response
            sr.Close();
            Response.Close();
            string json_resp = System.Web.HttpUtility.UrlDecode(result);
            return json_resp;

        }
```

6. Delete, Add, Unsubscribe or Update contact profile using HTTP JSON Post Interface

This section described HTTP post request that use to delete, add, unsubscribe or update contact profile. As you can see in the section below the request command field defined the command that will be executed.

HTTP post request should be sent to the following URL with UTF-8 encoding in the HTTP post request body.

http://slng5.com/Api/ApiGeneralCmdJson.ashx

Or (if SSL is needed)
https://slng5.com/Api/ApiGeneralCmdJson.ashx

6.2 General command JSON format

Below JSON format described the general API JSON format for request and response call. As mentioned before the following JSON format should be sent in the request body.

Note that for each command the table in the field section defined which fields are required.

Request JSON:

```
{
      "Username": "xxxxxxxx",
      "Password": "xxxxxxxxxxxx",
      "Cmd": "DeleteContactCmd",
      "GroupGUIDs": [],
      "Mobile": "",
      "Email": "test_api1234@slng.co.il",
      "FirstName": null,
      "LastName": null,
      "Street": null,
      "City": null,
      "Country": null,
      "PostalCode": null,
       "Fax": null
}
```

Response JSON:

```
{
        "Pass": true or false,
        "Msg":"Staus description",
}
```

6.3 Fields Description Request

Please note that the number next to the field type (e.g: String(200)) defined the maximum string length of this field.

Field Name	Description	Type
Username	your API username for SLNG system	String
Password	your API password for SLNG system	String
Cmd	Valid command are: (1) DeleteContactCmd (2) UnsubscribeContactCmd (3) AddContactCmd (4) UpdateContactProfileCmd	String
GroupGUIDs	Define the target group for new contact, if not defined new contact will be added to the default group (optional for AddContactCmd and not needed for all other commands)	Array of string
Mobile	Contact mobile number (optional only if email address defined)	String
Email	Contact email address (optional only if mobile number defined)	String
FirstName	Contact first name	String (200)
LastName	Contact last name	String (200)
Street	Contact street name	String (200)

City	Contact city name	String (150)
Country	Contact country	String (150)
PostalCode	Contact postal code	String (20)
Fax	Contact fax number	String (50)

6.4 Fields Description Response

Field Name	Description	Type
Pass	Return true or false value indicate the command status	Boolean
Msg	Described the message status	String

6.5 Fields Required/Optional per command

Table below described required, optional or not needed for each request command.

Field Name	DeleteContactCmd	UnsubscribeContactCmd	AddContactCmd	UpdateContactProfileCmd
Username	required	required	required	required
Password	required	required	required	required
Cmd	required	required	required	required
GroupGUIDs	not needed	not needed	optional	not needed
Mobile	required (email or mobile required)	required (email or mobile required)	required (email or mobile required)	required (email or mobile required)

Email	required (email or mobile required)	required (email or mobile required)	required (email or mobile required)	required (email or mobile required)
FirstName	not needed	not needed	optional	optional
LastName	not needed	not needed	optional	optional
Street	not needed	not needed	optional	optional
City	not needed	not needed	optional	optional
Country	not needed	not needed	optional	optional
PostalCode	not needed	not needed	optional	optional
Fax	not needed	not needed	optional	optional

6.6 Example for executing special command (delete contact) using HTTP post

6.6.1 HTTP Post – PHP example

```
<?php
function SendEmailJson ()
{
  $url = " http://slng5.com/Api/ApiGeneralCmdJson.ashx";
  $json ='{
      "Username": "xxxxxxxx",
      "Password": "xxxxxxxxxxxx",
      "Cmd": "DeleteContactCmd",
      "Email": "test_api1234@slng.co.il"

}';
     //-----------------------------------------------
     $CR = curl_init();
     curl_setopt($CR, CURLOPT_URL, $url);
     curl_setopt($CR, CURLOPT_POST, 1);
     curl_setopt($CR, CURLOPT_FAILONERROR, true);
     curl_setopt($CR, CURLOPT_POSTFIELDS, $json);
     curl_setopt($CR, CURLOPT_RETURNTRANSFER, 1);
     curl_setopt($CR, CURLOPT_HTTPHEADER,array("charset=utf-8"));
     //-----------------------------------------------
     $result = curl_exec($CR);
     $error = curl_error ($CR);
     $response = new SimpleXMLElement($result);
}
TestApiJson();

?>
```

6.6.2 HTTP Post – C# example

Example below use Json.NET library to Serialize and De-serialize JSON objects.

Json.NET library can be downloaded from the following link:

http://www.newtonsoft.com/json

Further instruction about the installation of the package/library in Visual Studio environment can be found here:
https://docs.microsoft.com/en-us/nuget/tools/package-manager-console

```
using Newtonsoft.Json;// Json.NET library

   public static bool TestApiGeneralCmdJson()
        {

            ApiGeneralCmdReqJSON obj = new ApiGeneralCmdReqJSON();
            ApiGeneralCmdResponseJSON obj1 = new ApiGeneralCmdResponseJSON();
            obj.Username = "xxxxxxx";
            obj.Password = "xxxxxxxxx";
            obj.Email = "test_api1234@slng.co.il";
            obj.Mobile = "";
            obj.FirstName = "API FIRST NAME TEST";
            obj.LastName = "API LAST NAME TEST";
            obj.Cmd = "DeleteContactCmd";
            string json = JsonConvert.SerializeObject(obj, Formatting.Indented);

            string json_enc = System.Web.HttpUtility.UrlEncode(json.ToString(),
System.Text.Encoding.UTF8);
            string json_resp;
            json_resp =
PostJsonBodyDataToSLNG("http://slng5.com/Api/ApiGeneralCmdJson.ashx", json_enc);

            obj1 =
JsonConvert.DeserializeObject<ApiGeneralCmdResponseJSON>(json_resp);
            if (obj1.Pass)
                return true;
            else
                return false;

        }



      internal class ApiGeneralCmdResponseJSON
        {
            private bool _Pass;
            public bool Pass { get { return _Pass; } set { _Pass = value; } }
            private string _Msg;
            public string Msg { get { return _Msg; } set { _Msg = value; } }

            public ApiGeneralCmdResponseJSON()
            {
                this.Pass = false;
                this._Msg = null;
            }
        }

        internal class ApiGeneralCmdReqJSON
        {
            private string _Username;
            public string Username { get { return _Username; } set { _Username =
value; } }

            private string _Password;
            public string Password { get { return _Password; } set { _Password =
value; } }

            private string _Cmd;
            public string Cmd { get { return _Cmd; } set { _Cmd = value; } }

            private List<string> _GroupGUIDs;
            public List<string> GroupGUIDs { get { return _GroupGUIDs; } set {
_GroupGUIDs = value; } }

            private string _Mobile;
            public string Mobile { get { return _Mobile; } set { _Mobile = value; } }

            private string _Email;
            public string Email { get { return _Email; } set { _Email = value; } }

            private string _FirstName;
            public string FirstName { get { return _FirstName; } set { _FirstName =
value; } }

            private string _LastName;
            public string LastName { get { return _LastName; } set { _LastName =
value; } }

            private string _Street;
            public string Street { get { return _Street; } set { _Street = value; } }

            private string _City;
            public string City { get { return _City; } set { _City = value; } }

            private string _Country;
            public string Country { get { return _Country; } set { _Country = value; }
}

            private string _PostalCode;
            public string PostalCode { get { return _PostalCode; } set { _PostalCode =
value; } }

            private string _Fax;
            public string Fax { get { return _Fax; } set { _Fax = value; } }
            public ApiGeneralCmdReqJSON()
            {
                this._Username = null;
                this._Password = null;
                this._Cmd = null;
                this.Email = null;
                this.Mobile = null;
                this.FirstName = null;
                this.LastName = null;
                this.City = null;
                this.Country = null;
                this.Fax = null;
                this.PostalCode = null;
                this.Street = null;
                this.GroupGUIDs = new List<string>();
            }
        }



     public static string PostJsonBodyDataToSLNG(string url, string json)
        {

            //Setup the web request
            string result = string.Empty;
            WebRequest Request = WebRequest.Create(url);
            Request.Timeout = 30000;
            Request.Method = "POST";
            Request.ContentType = "application/x-www-form-urlencoded";
            //Set the POST data in a buffer
            byte[] xml_encoding;

            json = json.Replace(" ", "+");
            //Specify the length of the buffer
            xml_encoding = Encoding.UTF8.GetBytes(json);
            Request.ContentLength = xml_encoding.Length;
            //Open up a request stream
            Stream RequestStream = Request.GetRequestStream();
            //Write the POST data
            RequestStream.Write(xml_encoding, 0, xml_encoding.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
            result = sr.ReadToEnd();
            //Close the reader, and response
            sr.Close();
            Response.Close();
            string json_resp = System.Web.HttpUtility.UrlDecode(result);
            return json_resp;

        }
```

7. Add, Deleted or Edit Group using HTTP JSON Post Interface

This section described HTTP post request that use to delete, add or edit groups. As you can see in the section below the request command field defined the command that will be executed.

HTTP post request should be sent to the following URL with UTF-8 encoding in the HTTP post request body.

http://slng5.com/Api/ApiGeneralCmdJson.ashx

Or (if SSL is needed)
https://slng5.com/Api/ApiGeneralCmdJson.ashx

8.2 General command JSON format

Below JSON format described the general API JSON format for request and response call. As mentioned before the following JSON format should be sent in the request body.

Note that for each command the table in the field section defined which fields are required.

Request JSON:

```
{
      "Username": "xxxxxxxx",
      "Password": "xxxxxxxxxxxx",
      "Cmd": " AddNewGroup",
      "GroupGUIDs": [],
      "GroupName": "New group name"
}
```

Response JSON:

```
{
        "Pass": true or false,
        "Msg":"Staus description"
}
```

8.3 Fields Description Request

Please note that the number next to the field type (e.g: String(200)) defined the maximum string length of this field.

Field Name	Description	Type
Username	your API username for SLNG system	String
Password	your API password for SLNG system	String
Cmd	Valid command are: (1) AddNewGroup (2) DeleteGroup (3) EditGroupName	String
GroupGUIDs	Define the group GUID field for editing or deleting group. (optional for AddNewGroup). Note that only the first item in the array is valid and there is no option for example to dlete multi groups in one command.	Array of string
GroupName	Define the group name filed needed for edit group name command and add new group command.	String (100)

8.4 Fields Description Response

Field Name	Description	Type
Pass	Return true or false value indicate the command status	Boolean
Msg	Described the message status	String

8.5 Fields Required/Optional per command

Table below described required, optional or not needed for each request command.

Field Name	AddNewGroup	DeleteGroup	EditGroupName
Username	required	required	required
Password	required	required	required
Cmd	required	required	required

GroupGUIDs	not needed	required	required
GroupName	required	Not needed	required

8.6 Example for executing add new group command using HTTP post

8.6.1 HTTP Post – PHP example

```
<?php
function SendEmailJson ()
{
  $url = " http://slng5.com/Api/ApiGeneralCmdJson.ashx";
  $json ='{
      "Username": "xxxxxxxx",
      "Password": "xxxxxxxxxxxx",
      "Cmd": " AddNewGroup",
      "GroupName": "NEW GROUP FROM API"

}';
     //-------------------------------------------------
     $CR = curl_init();
     curl_setopt($CR, CURLOPT_URL, $url);
     curl_setopt($CR, CURLOPT_POST, 1);
     curl_setopt($CR, CURLOPT_FAILONERROR, true);
     curl_setopt($CR, CURLOPT_POSTFIELDS, $json);
     curl_setopt($CR, CURLOPT_RETURNTRANSFER, 1);
     curl_setopt($CR, CURLOPT_HTTPHEADER,array("charset=utf-8"));
     //-------------------------------------------------
     $result = curl_exec($CR);
     $error = curl_error ($CR);
     $response = new SimpleXMLElement($result);
}
TestApiJson();

?>
```

8.6.2 HTTP Post – C# example

Example below use Json.NET library to Serialize and De-serialize JSON objects.

Json.NET library can be downloaded from the following link:

http://www.newtonsoft.com/json

Further instruction about the installation of the package/library in Visual Studio environment can be found here:
https://docs.microsoft.com/en-us/nuget/tools/package-manager-console

```
using Newtonsoft.Json;// Json.NET library

   public static bool TestApiAddGroupGeneralCmdJson()
        {

            ApiGeneralGroupCmdReqJSON obj = new ApiGeneralGroupCmdReqJSON();
            ApiGeneralCmdResponseJSON obj1 = new ApiGeneralCmdResponseJSON();
            obj.Username = "xxxxxxx";
            obj.Password = "xxxxxxxxx";
            obj.GroupName = "NEW GROUP FROM API";
            obj.Cmd = "AddNewGroup";
            string json = JsonConvert.SerializeObject(obj, Formatting.Indented);

            string json_enc = System.Web.HttpUtility.UrlEncode(json.ToString(), 
System.Text.Encoding.UTF8);
            string json_resp;
            json_resp = 
PostJsonBodyDataToSLNG("http://slng5.com/Api/ApiGeneralCmdJson.ashx", json_enc);

            obj1 = 
JsonConvert.DeserializeObject<ApiGeneralCmdResponseJSON>(json_resp);
            if (obj1.Pass)
                return true;
            else
                return false;

        }

      internal class ApiGeneralCmdResponseJSON
        {
            private bool _Pass;
            public bool Pass { get { return _Pass; } set { _Pass = value; } }

            private string _Msg;
            public string Msg { get { return _Msg; } set { _Msg = value; } }

            public ApiGeneralCmdResponseJSON()
            {
                this.Pass = false;
                this._Msg = null;
            }
        }


    internal class ApiGeneralGroupCmdReqJSON
        {
            private string _Username;
            public string Username { get { return _Username; } set { _Username = 
value; } }

            private string _Password;
            public string Password { get { return _Password; } set { _Password = 
value; } }

            private string _Cmd;
            public string Cmd { get { return _Cmd; } set { _Cmd = value; } }
            private List<string> _GroupGUIDs;
            public List<string> GroupGUIDs { get { return _GroupGUIDs; } set { 
_GroupGUIDs = value; } }
            private string _GroupName;
            public string GroupName { get { return _GroupName; } set { _GroupName = 
value; } }

            public ApiGeneralGroupCmdReqJSON()
            {
                this._Username = null;
                this._Password = null;
                this._Cmd = null;
                this.GroupGUIDs = new List<string>();
                this.GroupName = null;
            }
        }
    public static string PostJsonBodyDataToSLNG(string url, string json)
        {

            //Setup the web request
            string result = string.Empty;
            WebRequest Request = WebRequest.Create(url);
            Request.Timeout = 30000;
            Request.Method = "POST";
            Request.ContentType = "application/x-www-form-urlencoded";
            //Set the POST data in a buffer
            byte[] xml_encoding;
            json = json.Replace(" ", "+");
            //Specify the length of the buffer
            xml_encoding = Encoding.UTF8.GetBytes(json);
            Request.ContentLength = xml_encoding.Length;
            //Open up a request stream
            Stream RequestStream = Request.GetRequestStream();
            //Write the POST data
            RequestStream.Write(xml_encoding, 0, xml_encoding.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
            result = sr.ReadToEnd();
            //Close the reader, and response
            sr.Close();
            Response.Close();
            string json_resp = System.Web.HttpUtility.UrlDecode(result);
            return json_resp;

        }
```

8. Cancel Email/Sms Sending command using HTTP JSON Post Interface

This section described HTTP post request that allow to cancel sending command that already sent and schedule for future time.

HTTP post request should be sent to the following URL with UTF-8 encoding in the HTTP post request body.

http://slng5.com/Api/ApiGeneralCmdJson.ashx

Or (if SSL is needed)
https://slng5.com/Api/ApiGeneralCmdJson.ashx

10.2 General command JSON format

Below JSON format described the general API JSON format for request and response call. As mentioned before the following JSON format should be sent in the request body.

Note that for each command the table in the field section defined which fields are required.

Request JSON:

```
{
      "Username": "xxxxxxxx",
      "Password": "xxxxxxxxxxxx",
      "Cmd": "CancelSendingCommand",
      "MsgInfoGUID": "(received from the sending command)"
}
```

Response JSON:

```
{
        "Pass": true or false,
        "Msg":"Staus description"
}
```

10.3 Fields Description Request

Field Name	Description	Type
Username	your API username for SLNG system	String

Password	your API password for SLNG system	String
Cmd	Valid command are: (1) CancelSendingCommand	String
MsgInfoGUID	Define the MsgInfoGUID field that rerun from the sending command.	String

10.4 Fields Description Response

Field Name	Description	Type
Pass	Return true or false value indicate the command status	Boolean
Msg	Described the message status	String

10.5 Fields Required/Optional per command

Table below described required, optional or not needed for each request command.

Field Name	CancelSendingCommand
Username	required
Password	required
Cmd	required
MsgInfoGUID	required

9. Get user account balance using HTTP JSON Post Interface

This section described the HTTP post request for getting the user account balance.

HTTP post request should be sent to the following URL with UTF-8 encoding in the HTTP post request body.

http://slng5.com/Api/GetUserBalanceJson.ashx

Or (if SSL is needed)
https://slng5.com/Api/GetUserBalanceJson.ashx

12.2 Get user balance command JSON format

Below JSON format described the general API JSON format for get user balance command. As mentioned before the following JSON format should be sent in the request body.

Request JSON:

```
{
      "Username": "xxxxxxxx",
      "Password": "xxxxxxxxxxxx",
}
```

Response JSON:

```
{
        "Pass": true or false,
        "Msg":"Staus description",
        "QtySms": "100",
        "QtyEmail":"100"",
}
```

12.3 Fields Description Request

.

Field Name	Description	Type
Username	your API username for SLNG system	String
Password	your API password for SLNG system	String

12.4 Fields Description Response

Field Name	Description	Type
Pass	Return true or false value indicate the command status	Boolean
Msg	Described the message status	String
QtySms	Total number of available SMS in the account	String
QtyEmail	Total number of available email in the account	String

12.5 Example for executing special command (delete contact) using HTTP post

12.5.1 HTTP Post – PHP example

```
<?php
function SendEmailJson ()
{
  $url = " http://slng5.com/Api/ GetUserBalanceJson.ashx";
  $json ='{
      "Username": "xxxxxxxx",
      "Password": "xxxxxxxxxxxx",


}';
     //-----------------------------------------------
     $CR = curl_init();
     curl_setopt($CR, CURLOPT_URL, $url);
     curl_setopt($CR, CURLOPT_POST, 1);
     curl_setopt($CR, CURLOPT_FAILONERROR, true);
     curl_setopt($CR, CURLOPT_POSTFIELDS, $json);
     curl_setopt($CR, CURLOPT_RETURNTRANSFER, 1);
     curl_setopt($CR, CURLOPT_HTTPHEADER,array("charset=utf-8"));
     //-----------------------------------------------
     $result = curl_exec($CR);
     $error = curl_error ($CR);
     $response = new SimpleXMLElement($result);
}
TestApiJson();

?>
```

12.5.2 HTTP Post – C# example

Example below use Json.NET library to Serialize and De-serialize JSON objects.

Json.NET library can be downloaded from the following link:

http://www.newtonsoft.com/json

Further instruction about the installation of the package/library in Visual Studio environment can be found here:
https://docs.microsoft.com/en-us/nuget/tools/package-manager-console

```
using Newtonsoft.Json;// Json.NET library

     public static bool TestGetUserBalanceApiJson()
        {

            UserBalanceCmdJson obj = new UserBalanceCmdJson();
            UserBalanceResponseJson obj1 = new UserBalanceResponseJson();
            obj.Username = "xxxxx";
            obj.Password = "xxxxxxx";

            string json = JsonConvert.SerializeObject(obj, Formatting.Indented);

            string json_enc = System.Web.HttpUtility.UrlEncode(json.ToString(),
System.Text.Encoding.UTF8);
            string json_resp;
            json_resp =
PostJsonBodyDataToSLNG("http://slng5.com/Api/GetUserBalanceJson.ashx",  json_enc);

            obj1 = JsonConvert.DeserializeObject<UserBalanceResponseJson>(json_resp);
            if (obj1.Pass)
                return true;
            else
                return false;

        }


   internal class UserBalanceCmdJson
        {
            private string _Username;
            public string Username { get { return _Username; } set { _Username =
value; } }

            private string _Password;
            public string Password { get { return _Password; } set { _Password =
value; } }


            public UserBalanceCmdJson()
```

```
      {
        this.Password = null;
        this.Username = null;
      }
    }


    internal class UserBalanceResponseJson
    {
      private bool _Pass;
      public bool Pass { get { return _Pass; } set { _Pass = value; } }

      private string _Msg;
      public string Msg { get { return _Msg; } set { _Msg = value; } }

      private string _QtyEmail;
      public string QtyEmail { get { return _QtyEmail; } set { _QtyEmail =
value; } }

      private string _QtySms;
      public string QtySms { get { return _QtySms; } set { _QtySms = value; } }


      public UserBalanceResponseJson()
      {
        this._Msg = null;
        this._Pass = false;
        this.QtyEmail = null;
        this.QtySms = null;
      }

    }




  public static string PostJsonBodyDataToSLNG(string url, string json)
    {

      //Setup the web request
      string result = string.Empty;
      WebRequest Request = WebRequest.Create(url);
      Request.Timeout = 30000;
      Request.Method = "POST";
      Request.ContentType = "application/x-www-form-urlencoded";
      //Set the POST data in a buffer
      byte[] xml_encoding;

      json = json.Replace(" ", "+");
      //Specify the length of the buffer
      xml_encoding = Encoding.UTF8.GetBytes(json);
      Request.ContentLength = xml_encoding.Length;
      //Open up a request stream
      Stream RequestStream = Request.GetRequestStream();
      //Write the POST data
      RequestStream.Write(xml_encoding, 0, xml_encoding.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
            result = sr.ReadToEnd();
            //Close the reader, and response
            sr.Close();
            Response.Close();
            string json_resp = System.Web.HttpUtility.UrlDecode(result);
            return json_resp;

        }
```

10. Sending SMS using Web Service

SLNG provide the option to send SMS messages using web service interface. In this document we described the basic capability of our API web service. If further support is needed, please contact SLNG support team.

List of all supported API command can be found here:

http://www.slng6.com/WebService/SendSms.asmx

For Microsoft .Net users:
Microsoft .Net users should add reference to this URL:
http://www.slng6.com/WebService/SendSms.asmx

Or download the DLL from the following path:
http://www.slng.co.il/Downloads/API/DLL/SLNG_SmsApi.dll

For Java / Delphi / Magic users:
Java / Delphi / Magic users should add reference to this URL:
http://www.slng6.com/WebService/SendSms.asmx?wsdl

13.3 Basic example for Sending SMS messages

In this example we described the basic web service method to send SMS message to multi contacts.

Name of the method: CreateAndSendSmsToMultiContactsByMobileNo

URL for web service:
http://www.slng6.com/WebService/SendSms.asmx?op=CreateAndSendSmsToMultiContactsByMobileNo

Following are the input fields to this method:

- Username – your API username for SLNG system
- Username – your API password for SLNG system
- MsgName - for internal use to find the message in our application
- GlobalID = "-1" (should be always set to -1)
- Msg - your message body
- FromMobile - sender number
- Mobiles - Destination number, one or more numbers separated by " , "

13.3.1 C# example

Below VB example is based on the available DLL for the web service.

```
private void SendSmsMsgByMobile()        {

      string username = "";
      string password = "";
```

```
        string FromMobile = null;
        string Mobiles = null;

        StatusMultiCmds status = new StatusMultiCmds();
        status = SendSmsApi.CreateAndSendSmsByMobileNo(username, password, "Msg name
here", "this is the actual message body", FromMobile, Mobiles);

}
```

13.3.2 VB example

Below VB example is based on the available DLL for the web service.

```
Private Sub SendSmsMsgByMobile()

        Dim username As String = ""
        Dim password As String = ""
        Dim  FromMobile As String
        Dim  Mobiles As String

        Dim status As New StatusMultiCmds()

        status = SendSmsApi.CreateAndSendSmsByMobileNo(username, password, "Msg name here", "this
is the actual message body", FromMobile, Mobiles)

    End Sub
```

13.3.3 PHP Example

```
$postfields = array(
                        'Username' => '',
                        'Password' => '',
'MsgName'=> 'MSG NAME HERE',
'GlobalID'=> '-1',
'Msg'=> 'your SMS message is here',
'FromMobile'=> '',
'Mobiles'=> '0523243903, 052555555, 0546666666'
                   );

    // open a curl connection
     $ch = curl_init();
       curl_setopt($ch, CURLOPT_URL,
'http://www.slng6.com/WebService/SendSms.asmx/CreateAndSendSmsToMultiContactsByMobileNo?'.http_b
uild_query($postfields));

 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

 $result = curl_exec($ch);
```

13.4 Sending SMS message with client profile update

**** To simplify the deployment please talk with us before you use this method ****

In this section we described a sending SMS command that also update the customer profile. You can use this command for sending personalized SMS messages.

For an example to send message with first name, you need first to define the first name of the customer and also to include the first name tag in the message body in order to tell SLNG to replace the first name tag with the customer first name as described in the example below.

Msg = "Hello ##שם_פרטי##" - for this message the SLNG system will replace the first name of the customer with ##שם_פרטי## during the sending message process.

This command split into 3 commands as following:

1. Create SMS message
2. Add contact to message
3. Send message

Following are the input fields to this method:

- Username – your API username for SLNG system
- Password – your API password for SLNG system
- MsgName - for internal use to find the message in our application
- GlobalID = "-1" (should be always set to -1)
- Msg - your message body
- FromMobile - sender number

13.4.1 VB Example

Below VB example is based on the ASP>NET web service DLL.

```
'STEP 1 CREATE SMS MESSAHE
Dim username As String = ""
Dim password As String = "" '

Dim status As New StatusCmd
Dim MsgInfoGUID As String
Dim MsgBody As String = "This is the SMS message"
Dim MsgName As String = "This is my message name"
Dim Sender As String = "xxxxx"

status = New StatusCmd
status = SendSmsApi.CreateSmsMsg_W_GlobalID(username, password, MsgName ,  "-1",
MsgBody  ,  Sender  ,  Sender  )
```

```
 MsgInfoGUID = status.GeneralGUID


'STEP 2 ADDING CONTACTS TO MESSAGE
 Dim contact1 As New ApiContactInfo

 contact1.Email = xxxxx"
 contact1.FirstName = "Shlomi"
 contact1.LastName = "Levy"
 contact1.Mobile = "xxxxx"

 Dim contact2 As New ApiContactInfo
 contact2.Email = "xxxxx"
 contact2.FirstName = "Shlomi"
 contact2.LastName = "Levy"
 contact2.Mobile = "xxxxx"

 Dim contacts(1) As ApiContactInfo
 contacts(0) = (contact1)
 contacts(1) = (contact2)

 Dim status1 As New StatusMultiCmds
     status1 = SendSmsApi.AddMultiContactsToSmsMsg(username, password, MsgInfoGUID,
     contacts)

     'STEP3 SEND MSG'
     SendSmsApi.SendSmsMsg(username, password, MsgInfoGUID)
```

11. EmailToSms Interface

SLNG supports a simple format of Email to Sms (EmailToSms). We found that this interface simplify the integration between third party vendors and SLNG for system that require basic integration.

Section below will described the email format.

13.4.2 Sending message

With EmailToSms interface you can define the following:

- Email address – define the destination mobile number with the following format:
  To: [mobile_number]@slng6.com
  You can define up to 500 numbers per message
- Subject line should follow the following format:
  subject: username##password##from mobile##schedule time
  - Username – SLNG API username
  - Password - SLNG API password

  - From mobile – Optional field, defines the sender number or name in English up to 11 characters. If not define, the value will be taken from the sender profiler.
  - Schedule time – Optional field, defines the schedule time for sending message. Note: if schedule time field defines
- Message body – Design the actual SMS message body. Note that email need to be sent in TXT format and not in HTML format.

Few notes:

1. Email should be sent in TXT format and not in HTML format.
2. From mobile is an optional filed, however if schedule time defines the from mobile field place folder must be used with empty string as you can see in the example below

13.4.3 EmailToSms – Example

Example 1: Basic example:

In the basic example below the from mobile (sender) will be taken from the profiler and the message schedule to be sent immediately with no delay (schedule time has not been defined)

- Email: In TX format
- To: 052-9999999@slng6.com; 052-44444444@slng6.com
- Subject: username##password

Example 2: Schedule message

In the example below we schedule the message for future sending and the from mobile field (sender) is taken from the sender profiler

- Email: In TX format
- To: 052-9999999@slng6.com; 052-44444444@slng6.com
- Subject: username##password####26/11/2018 14:25

Example 2: Schedule message + from mobile

In the example below we schedule the message for future sending and the rom mobile field also define in the command line.

- Email: In TX format
- To: 052-9999999@slng6.com; 052-44444444@slng6.com
- Subject: username##password##055-3333333##26/11/2018 14:25

12. User Registration Interface

SLNG supports to method for user registration:

1. Email registration interface allows you to implement a simple integration between SLNG and third party tools by sending email command for register new contacts.
2. HTTP GET call for user registration

All the commands below are full supported by web service interface, however in some cases, it is easier to use the above method to implement a simple integration between third party software and SLNG.

16.7 Email to Registration - Command

Below you can find the email structure to send email to registration command

- Email address – use as identifier for the command and must be set to:
- To: 0000000@slng6.com
- Subject line should follow the following format and define the following fields:
  subject: username##password##GroupGUID##FirstName##LastName##Email##Mobile
  - Username – SLNG API username
  - Password - SLNG API password
  - GroupGUID – Define the target group GUID field (can be taken from the URL of the group contact list view)
  - FirstName – Optional field, defines the contact first name, if not defined it should be empty
  - LastName – Optional field, defines the contact last name, if not defined it should be empty
  - Email – Optional field, defines the contact email, if not defined it should be empty. Note either field, email or mobile, must be defined.
  - Mobile – Optional field, defines the contact mobile number, if not defined it should be empty. Note either field, email or mobile, must be defined.

16.7.1 EmailToReg - Examples

Basic Example - Email

Define the contact email address only. GroupGUID filed value should be taken from the system.

- To: 0000000@slng6.com
- Subject: username##password##GroupGUID######test@slng.co.il##

Basic Example - Mobile

Define the contact email address only. GroupGUID filed value should be taken from the system.

- To: 0000000@slng6.com
- Subject: username##password##GroupGUID########052-5555555

Example with more fields

Define the contact email address only. GroupGUID filed value should be taken from the system.

- To: 0000000@slng6.com
- Subject: username##password##GroupGUID##Gil##Levy##test@slng.co.il##052-5555555

16.8 User registration via HTTP post

SLNG support basic new contact registration via HTTP post.

Customer can easily generate HTTP GET request as described below for simple user registration. Note that customer can define redirect URL if needed and it can be sued for example to redirect the new contact to thank you or welcome page at the end of the registration if needed.

Field description:

- Username - customer API username
- Password - Customer API password
- Email - user email address (optional, but if not defined user need to define the mobile number)
- Mobile - user mobile number (optional, but if not defined the user need to define the email)
- FirstName - user first name (optional)
- Last Name - user last name (optional)
- RedirectURL - define the redirect address. if defined, new contact will redirected to the new page at the end of the registration even if the registration fail (optional)
- GroupGUID- define the new contact target group GUID file, mandatory for AddContactToGroup command.
- Cmd - Command to execute, supported command are the following: AddContactToGroup , AddNewContact

Example:

http://www.slng5.com/Api/SlngApi.ashx?Username=xxxxxxxxxxx&password=xxxxxx&email=xxxxx@slng.co.il&mobile=xxxxxxxxxxx&cmd=AddContactToGroup&GroupGUID=f040999e-47a5-4a75-b30a-711fca7e2688&redirecturl=http://www.ynet.co.il