

# Quick Start

## SLNG Basic API Instruction for Email Version - 5.6

## Contents

1.	Introduction .....	4
2.	Sending Email using HTTP JSON Post Interface .....	5
2.2	Sending Email JSON format.....	5
2.3	Fields Description Request.....	6
2.4	Fields Description Response .....	9
2.5	Sending Personalized Email message .....	9
2.6	Example for sending Email using HTTP post .....	10
2.6.1	HTTP Post – PHP example .....	10
2.6.2	HTTP Post – C# example.....	10
3.	Get Message statistics using HTTP JSON Post Interface .....	16
4.2	Get message statistics JSON format .....	16
4.3	Fields Description Request.....	17
4.4	Fields Description Response .....	17
4.5	Example for getting message statistics using HTTP post .....	18
4.5.1	HTTP Post – PHP example .....	18
4.5.2	HTTP Post – C# example.....	18
4.	Delete, Add, Unsubscribe or Update contact profile using HTTP JSON Post Interface .....	22
6.2	General command JSON format .....	22
6.3	Fields Description Request.....	23
6.4	Fields Description Response .....	24
6.5	Fields Required/Optional per command.....	24
6.6	Example for executing special command (delete contact) using HTTP post .....	25
6.6.1	HTTP Post – PHP example .....	25
6.6.2	HTTP Post – C# example.....	26
5.	Add, Deleted or Edit Group using HTTP JSON Post Interface .....	29
8.2	General command JSON format .....	29
8.3	Fields Description Request.....	29
8.4	Fields Description Response .....	30
8.5	Fields Required/Optional per command.....	30



8.6	Example for executing add new group command using HTTP post .....	31
8.6.1	HTTP Post – PHP example .....	31
8.6.2	HTTP Post – C# example.....	31
6.	Cancel Email/Sms Sending command using HTTP JSON Post Interface .....	34
10.2	General command JSON format .....	34
10.3	Fields Description Request.....	34
10.4	Fields Description Response .....	35
10.5	Fields Required/Optional per command.....	35
7.	Get user account balance using HTTP JSON Post Interface .....	37
12.2	Get user balance command JSON format.....	37
12.3	Fields Description Request.....	37
12.4	Fields Description Response .....	38
12.5	Example for executing special command (delete contact) using HTTP post .....	38
12.5.1	HTTP Post – PHP example .....	38
12.5.2	HTTP Post – C# example.....	39

## 1. Introduction

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

A separate document that can be [found here](#) described the SMS API.

**This document does not cover the complete API functionality but provide the most common use functionality, for further information please contact SLNG support team.**

Also please note that this document only described the JSON API that we recommend to use and as mentioned before for XML or web service interface please contact SLNG support team.

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

### **Http post Interface:**

1. Sending Email using JSON HTTP post interface
2. Get message statistics using JSON HTTP post interface.
3. Unsubscribe contact using JSON HTTP POST interface.
4. Delete contact using JSON HTTP post interface.
5. Add new contact using JSON HTTP post interface.
6. Update contact profile using HTTP post interface
7. Get user account balance using HTTP post interface.

## 2. Sending Email 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/SendEmailJson.ashx>

Or (if SSL is needed)

<https://slng5.com/Api/SendEmailJson.ashx>

### 2.2 Sending Email JSON format

Below JSON format described the sending command. As mentioned before the following JSON format should be sent in the request body of the HTTP call.

Note that the HTML request must be sent with html and body tag.

#### **Request:**

```
{
  "Username": "your API username",
  "Password": "your API password",
  "MsgName": "Your message name",
  "MsgBody": "Message body here",
  "FromName": "Sender name",
  "FromEmail": "Sender Email",
  "ReplyToEmail": "Reply to email",
  "MsgBodyType": "message body type("1" for HTML format)",
  "LanguageType": "Language type ("1" for Hebrew)",
  "Subject": "Email subject",
  "MsgScheduleTime": "message schedule time (optional)",
  "GroupGUIDs": [
    "Array of target group (GroupGUID) for this message",
  ],
  "Emails": [
    {
      "Email": "contact email",
      "FirstName": "Contact first name",
      "LastName": "Contact last name",
      "Street": "Contact Street",
      "City": "Contact city",
      "Country": "Contact country",
      "PostalCode": "Contact postal code",
      "Fax": "Contact fax"
    }
  ]
}
```

**Response:**

Note that GeneralGUID field defines the MsgInfoGUID of the message and it use by some other command (for example, MsgInfoGUID field use to get the message statistics).

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

### 2.3 Fields Description Request

Field Name	Description	Type	Required/Optional
<b>Username</b>	your API username for SLNG system	String	Required
<b>Password</b>	your API password for SLNG system	String	Required
<b>MsgName</b>	Defines the message name that will present in SLNG system.	String	Required
<b>MsgBody</b>	Your email message body. If you send HTML you HTML message has to include <html> and <body> tags.	String	Required
<b>FromName</b>	Defines the email from name field.	String	Required
<b>FromEmail</b>	Defines the email from email field.	String	Required
<b>ReplyToEmail</b>	Define the reply to email fileld.	String	Required
<b>MsgBodyType</b>	Define the email message body type format: "1" – HTML format "2" – Text format	String	Required
<b>LanguageType</b>	Define the email message language: "1" – Hebrew "2" - English "3" – Russia "4" - Arabic	String	Required
<b>Subject</b>	Define the email message subject	String	Required



Email Marketing & SMS Solution

<b>GroupGUIDs</b>	Define the target GroupGUID in SLNG for this message.	Array of strings	Optional only if Emails field below defined
<b>Emails</b>	Array of email object that defines the contact profile fields. For more information see description of email object below	Array of Email object	Optional only if GroupGUIDs field defined

### Email Objet:

Email object defined in the Emails fields of the message, as defined in the above field description table this object is optional if the GroupGUIDs field is defined.

Please be aware of the following:

1. The contact object fields can be used to create a personalized email message as described in the next section.
2. The number next to the field type (e.g: String(200)) defined the maximum string length of this field.

Field Name	Description	Type	Required/Optional
<b>Email</b>	Contact email address. Note that this field is required only if the email object defined.	String	Required
<b>FirstName</b>	Contact first name field.	String (200)	Optional
<b>LastName</b>	Contact last name field.	String (200)	Optional
<b>Street</b>	Contact street field.	String (200)	Optional
<b>City</b>	Contact city field	String (150)	Optional
<b>Country</b>	Contact country field.	String (150)	Optional
<b>PostalCode</b>	Contact postal code field.	String (20)	Optional
<b>Fax</b>	Contact Fax field.	String (50)	Optional



## 2.4 Fields Description Response

Field Name	Description	Type
<b>Status</b>	Pass or Fail – defines if the sending email command pass or fail.	Boolean
<b>Description</b>	Status description message	String
<b>GeneralGUID</b>	Carries the message MsgInfoGUID field that use to get the message statistics	String

## 2.5 Sending Personalized Email message

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

Personalized fields can be defined in the email message body or in the subject.

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 ##פרטי_שם##
2	LastName	200	##LAST_NAME## OR ##משפחה_שם##
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 for sending Email using HTTP post

### 2.6.1 HTTP Post – PHP example

```
<?php
function SendEmailJson ()
{
    $url = " http://sln5.com/Api/SendEmailJson.ashx ";
    $json = '{
        "Username": "xxxxxxx",
        "Password": "xxxxxxx",
        "MsgName": "Your message name",
        "MsgBody": addslashes("<html><head><title></title></head><body dir=\"rtl\"><p>this
is a test</p></body></html ")",
        "FromName": "SLNG Support",
        "FromEmail": "test@test.co.il",
        "ReplyToEmail": " test@test.co.il",
        "MsgBodyType": "1",
        "LanguageType": "1",
        "Subject": "Email subject",

        "Emails": [
            {
                "Email": "test@test.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);
    $response = json_decode(urldecode($result));

}
TestApiJson();

?>
```

### 2.6.2 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 TestEmailApiJson()
{
    string msg =
System.Security.SecurityElement.Escape("<html><head><title></title></head><body
dir=\"rtl\"><p>this is a test</p></body></html>");

    List<string> destEmail = new List<string>();
    destEmail.Add("support@slnG.co.il");
    destEmail.Add("shlomo@slnG.co.il");

    SendEmailCmdJson obj = new SendEmailCmdJson();
    obj.Username = "xxxxxxx";
    obj.Password = "xxxxxxx";
    obj.MsgName = "Test SLNG API";
    obj.MsgBody = msg;
    obj.FromEmail = "support@slnG.co.il";
    obj.FromName = "SLNG Support";
    obj.LanguageType = "1";
    obj.MsgBodyType = "1";
    obj.ReplyToEmail = "suppoer@slnG.co.il";
    obj.Subject = "Test Email";

    obj.DeliveryAckUrl = null;
    obj.HighPriority = false;
    foreach (string email in destEmail)
    {
        EmailInfoJson obj1 = new EmailInfoJson();
        obj1.Email = email;
        obj1.FirstName = "Shlomi";
        obj1.LastName = "Haddad";
        obj1.Emails.Add(obj1);
    }

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

    string json_enc = System.Web.HttpUtility.UrlEncode(json.ToString(),
System.Text.Encoding.UTF8);
    EmailResponseJson obj2 = new EmailResponseJson();
    obj2 = PostEmailJsonDataToSLNG ("http://slnG5.com/Api/SendEmailJson.ashx",
"SlnGJson=" + json_enc);

    if (obj2.Status)
        return true;
    else
        return false;
}
```

```

}

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

    private string _AckID;
    public string AckID { get { return _AckID; } set { _AckID = 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 EmailInfoJson()
    {
        this.AckID = null;
        this._Email = null;
        this.FirstName = null;
        this.LastName = null;
        this.City = null;
        this.Country = null;
        this.Fax = null;
        this.PostalCode = null;
        this.Street = null;
    }
}

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

    private string _Password;

```



Email Marketing & SMS Solution

```
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 _FromName;
public string FromName { get { return _FromName; } set { _FromName =
value; } }

private string _FromEmail;
public string FromEmail { get { return _FromEmail; } set { _FromEmail =
value; } }

private string _ReplyToEmail;
public string ReplyToEmail { get { return _ReplyToEmail; } set {
_ReplyToEmail = value; } }

private string _MsgBodyType;
public string MsgBodyType { get { return _MsgBodyType; } set {
_MsgBodyType = value; } }

private string _LanguageType;
public string LanguageType { get { return _LanguageType; } set {
_LanguageType = value; } }

private string _Subject;
public string Subject { get { return _Subject; } set { _Subject = value; }
}

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

private bool _HighPriority;
public bool HighPriority { get { return _HighPriority; } set {
_HighPriority = value; } }

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

private List<EmailInfoJson> _Emails;
public List<EmailInfoJson> Emails { get { return _Emails; } set { _Emails
= value; } }

public SendEmailCmdJson()
{
    this.DeliveryAckUrl = null;
    this.FromEmail = null;
    this.FromName = null;
    this.LanguageType = null;
    this.MsgBodyType = null;
}
```

```

        this.ReplyToEmail = null;
        this._Subject = null;
        this.Emails = new List<EmailInfoJson>();
        this.MsgBody = null;
        this.MsgName = null;
        this.Password = null;
        this.Username = null;
        this.HighPriority = false;
        this.GroupGUIDs = new List<string>();
    }
}

public class EmailResponseJson
{
    private bool _Status;
    public bool Status { get { return _Status; } set { _Status = value; } }

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

    private string _GeneralGUID;
    public string GeneralGUID { get { return _GeneralGUID; } set {
_GeneralGUID = value; } }

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

public static EmailResponseJson PostEmailJsonDataToSLNG(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<EmailResponseJson>(json_resp); ;
}
```

### 3. 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
<b>Username</b>	your API username for SLNG system	String	Required
<b>Password</b>	your API password for SLNG system	String	Required
<b>MsgInfoGUID</b>	MsgInfoGUID field return from the sending email command.	String	Required

### 4.4 Fields Description Response

Field Name	Description	Type
<b>Pass</b>	Return true or false value indicate the command status	Boolean
<b>Msg</b>	Described the message status	String
<b>TotalContact</b>	Total number of contacts that associated with this message	String
<b>TotalSent</b>	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
<b>TotalOpen</b>	Total contacts that open the message	String
<b>TotalError</b>	Total error emails in the list (hard error)	String
<b>TotalReturn</b>	Total email that sent but return (soft error)	String
<b>TotalUnsubscribe</b>	Total unsubscribed contacts.	String
<b>TotalClick</b>	Total clicks on the message	String
<b>TotalContactClick</b>	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()
    {
```



Email Marketing & SMS Solution

```
MsgStatisticsReqJson obj = new MsgStatisticsReqJson();
MsgStatisticsResponseJson obj1 = new MsgStatisticsResponseJson();
obj.Username = "xxxxxxx";
obj.Password = "xxxxxxx";
obj.MsgInfoGUID = "xxxxxxxxxxxxxxxxxxxx";
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;

}
```

## 4. 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
<b>Username</b>	your API username for SLNG system	String
<b>Password</b>	your API password for SLNG system	String
<b>Cmd</b>	Valid command are: (1) DeleteContactCmd (2) UnsubscribeContactCmd (3) AddContactCmd (4) UpdateContactProfileCmd	String
<b>GroupGUIDs</b>	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
<b>Mobile</b>	Contact mobile number (optional only if email address defined)	String
<b>Email</b>	Contact email address (optional only if mobile number defined)	String
<b>FirstName</b>	Contact first name	String (200)
<b>LastName</b>	Contact last name	String (200)
<b>Street</b>	Contact street name	String (200)

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

#### 6.4 Fields Description Response

Field Name	Description	Type
<b>Pass</b>	Return true or false value indicate the command status	Boolean
<b>Msg</b>	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
<b>Username</b>	required	required	required	required
<b>Password</b>	required	required	required	required
<b>Cmd</b>	required	required	required	required
<b>GroupGUIDs</b>	not needed	not needed	optional	not needed
<b>Mobile</b>	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": "xxxxxxx",
        "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@slnge.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://slnge5.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;
}

```

## 5. 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
<b>Username</b>	your API username for SLNG system	String
<b>Password</b>	your API password for SLNG system	String
<b>Cmd</b>	Valid command are: (1) AddNewGroup (2) DeleteGroup (3) EditGroupName	String
<b>GroupGUIDs</b>	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 delete multi groups in one command.	Array of string
<b>GroupName</b>	Define the group name field needed for edit group name command and add new group command.	String (100)

#### 8.4 Fields Description Response

Field Name	Description	Type
<b>Pass</b>	Return true or false value indicate the command status	Boolean
<b>Msg</b>	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
<b>Username</b>	required	required	required
<b>Password</b>	required	required	required
<b>Cmd</b>	required	required	required

<b>GroupGUIDs</b>	not needed	required	required
<b>GroupName</b>	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": "xxxxxxx",
        "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;
```





Email Marketing & SMS Solution

```
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;
}
```

## 6. 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": "xxxxxxx",
  "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

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

#### 10.4 Fields Description Response

Field Name	Description	Type
<b>Pass</b>	Return true or false value indicate the command status	Boolean
<b>Msg</b>	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
<b>Username</b>	required
<b>Password</b>	required
<b>Cmd</b>	required
<b>MsgInfoGUID</b>	required



## 7. 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": "xxxxxxxxxxxxxx",
}
```

#### **Response JSON:**

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

### 12.3 Fields Description Request

Field Name	Description	Type
<b>Username</b>	your API username for SLNG system	String
<b>Password</b>	your API password for SLNG system	String

## 12.4 Fields Description Response

Field Name	Description	Type
<b>Pass</b>	Return true or false value indicate the command status	Boolean
<b>Msg</b>	Described the message status	String
<b>QtySms</b>	Total number of available SMS in the account	String
<b>QtyEmail</b>	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": "xxxxxxxxxxxxx",

    }';
    //-----
    $CR = curl_init();
    curl_setopt($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;
}
```