Overview

This User Manual provides specifications of the HTTP based API provided by SpringEdge for the automated sending of SMS via the Internet. This guide is intended for developers and clients alike who plan to integrate their systems with SpringEdge'sMessaging service.

Required Environment

In order to use the HTTP based API, you will need to have a system that has internet connectivity. This connectivity may be direct to the internet or via a firewall or proxy server.

Additionally, should you wish to receive delivery reports (DLRs) from the SpringEdge system, you must have a web server that is able to receive incoming requests from the Internet. This web server does not necessarily have to be running on the same machine that is sending messages to SpringEdge.

This is one of the simpler server-based forms of communication. It can be used either in the form of a HTTP POST or HTTP GET. We recommend POST for larger data transfer and data security. All calls to the API must be URL-encoded. The parameter names are case sensitive.

HTTP Methods

HTTP POST or GET method is used in HTTP API for following things:

1. To send text messages: Your message and destination mobile numbers are sent to us. The message is then delivered to the destination mobile number by our system.
2. To receive delivery notifications: A notification with all required details are sent to a URL predetermined by you.
3. To check the message delivery status: The message delivery status is sent to you when you provide us with the unique message ID.
4. To check the available credits: you can check the available credits in your account.

Note: We recommend POST for larger data transfer and data security. All calls to the APImust be URL encoded.

SSL Enabled API Triggering

This API has been designed to allow you to access the API using an SSL Enabled connection for added security.

To trigger the API on an SSL enabled connection, enter the URL starting with https:// instead of http://

Input Data Formats

While making a request to the HTTP API, use form-encoded requests (just like submitting an HTML form). The default response for all requests will be in the plain text format. Response formats can be configured to JSON or XML by using an additional parameter.

Security

HTTP API supports both HTTP and HTTPS. Using HTTPS to encrypt all requests through SSL is recommended, but if you're using a client and tool that's not set up to make SSL requests, you can use HTTP. All requests are secured through BASIC authentication via the Working Key. An additional level of protection is provided where Working key can be configured to accept requests from a specific IP Address only.

Request Limitations

Maximum of 10 requests per second is allowed per IP address. Crossing the limitation by Sending huge requests from same IP Address (using unconditional loop) is prohibited. This could get your IP Address blocked temporarily. We recommend to use XML API instead using which different messages can be sent to different mobile numbers using single request.

Output Restrictions

If the provider network is down, our Gateway Servers will retry your message up to six hours before giving up.

Duplicate messages in single request are NOT filtered by default. A "duplicate message" is defined as the same destination address, sender address and message body.

Supported Character Sets

GSM 03.38 Encoding

The standard SMS mobile-phone character set is used universally to send text messages. The standard character set allows up to 160 characters, and includes few of the Roman alphabet and the most common accented forms, certain special characters, and few Greek alphabets. It includes the following characters:

@ £ \ $ ¥ è é ù ì ò Ç \ n Ø ø \ r Å å ? _ F " ? O ? ? S T ? Æ æ ß É ! \ " # ¤ % & ' ( ) * + , - ../ 0 1 2 3 4 5 6 7 8 9 : ; < = > ? ¡ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ? Ö Ñ Ü § ¿ a b c d e f g h I j k l m n o p q r s t u v w x y z ö ñ ü à ä Ä { } \ ~ [ ] |

Note: All extended characters will take two characters to send. Extended characters are:

£ ¥ è é ù ì ò Ç Ø ø Å å ΦΓΛΩΠΨΣΘΞ Æ æ ß É ¤ ¡ Ä Ö Ñ Ü § ¿ ä ö ñ ü à

Also, extended character like €, takes three characters to send.

Unicode

When using non-GSM 03.38 characters SMS transmits as a Unicode message. The Unicode character support provides the ability to send a message in Hindi, Kannada, Tamil etc.

Message Length - Multipart Messages

Standard GSM 03.38 Character Set

Single text messages are limited to 160 characters, including spaces. The maximum length of two-part or three-part multipart text messages is 306 (2 x 153) and 459 (3 x 153) characters, respectively. You could send up to 1000 characters as multipart message which will be split into 7 messages internally. This refers only to messages that contain the standard GSM 03.38 character set.

Each multipart text message is limited to 153 characters rather than 160 due to the need for user-data headers (UDHs) information. Mobile phones use UDH information to enable them to link long messages together so that they appear as single SMS messages in recipient's phone inboxes.

Number of SMS	Number of characters in the linked message
1	160 characters
2	306 characters (2 x 153)
3	459 characters (3 x 153)
...	...
7	1000 characters (6 x 153) + (1 x 82)

Unicode Characters

Using Unicode, for such languages as Hindi, restricts your message to a maximum of 70 characters per SMS. You can, however, send longer text messages, with our system splitting them into two or at the three separate text messages. The maximum lengths of two-part and three-part multipart Unicode text messages are 134 (2 x 67) and 201 (3 x 67) characters, respectively. You could send up to 500 characters as multipart message which will be split into 8 messages internally.

Number of SMS	Number of Characters in the linked message
1	70 characters
2	134 characters (2 x 67)
3	201 characters (3 x 67)
...	...
8	500 characters (7 x 67) + (1 x 31)

Mobile Phone Number Format

All mobile phone numbers submitted via API should be a valid 10 digit mobile number starting from 7, 8 or 9.
For example:

9xxxxxxxxx, 8xxxxxxxxx, 7xxxxxxxxx

Sender ID Options

All text messages carry a sender ID. When you send a text message through HTTP API, you could use any of the approved sender IDs for your account. You can reach us for applying for a new sender ID.

Setting a Sender ID (Alphabetic Originator)

You can set the sender to be your company name, or anything else up to a maximum of 6 characters. So, this means that your recipients could see the name you want in the sender field rather than just a phone number. This alphanumeric originator, or sender ID, can contain characters from a-z, A-Z. Other characters may cause messages to fail.

Sample HTTP API URL Format

The HTTP API can be used by clients to send messages to SpringEdge's server. The HTTP API URL is in the following format:

http://<SMS_Service_Url>/api/web/send/?apikey==XXXXXXXXXXXXXXXXX&sender=XXXXXX&to=9900XXXXXX&message=<HELLO>&format=json&dlrurl=<DLRURL>&custom=XXX

API Parameter Specifications

The API parameter specifications are described in the following sections.

Mandatory Parameters

The mandatory parameters in the HTTP API are tabulated below:

NAME	PARAMETER	DESCRIPTION	REQUIRED VALUE
URL	SMS_Service_URL	URL of your SMS Service	URL
API Key	Apikey	API key generated from your SMS Account	API Key
Sender ID	sender	Sender ID assigned to your account	Sender ID
Mobile Number	to	Mobile number to which the SMS is to be sent	Mobile Number
Message	message	Message to be sent	Message text which is URL encoded

Optional Parameters

The optional parameters in the HTTP API are tabulated below:

NAME	PARAMETER	DESCRIPTION	REQUIRED VALUE
Format	format	Output format should be as specified by this variable	XML/JSON

Successful Submission of SMS

For each successful submission of an SMS, the API would return a unique message ID for that transaction. The Client's Application should capture the unique message ID and fetch a Delivery Report (DLR) against that corresponding message ID.

This message ID then can be used to track the status of the message and receive the delivery report against this message ID through the delivery report URL specified by the client.

Response Formats

By default, the HTTP API URL generates responses in plain text format. You can also generate responses in JSON and XML formats using the API URL. All you need to do is add &text=json (for JSON format) or &text=xml (for XML format) at the end of the HTTP API URL.

JSON Format

For example, to receive the response in JSON format, the URL should be of the following format.

http://<SMS_Service_URL>/api/web/send?apikey= XXXXXXXXXXXXXXXX
&sender=XXXXXX&to=9900xxxxxx&message=TEST&format=json

The response in JSON format will be as shown below.

{"groupID":61,"MessageIDs":"61-1","status":"AWAITED-DLR"}

XML Format

To receive the response in XML format, the URL should be of the following format.

http://<SMS_Service_URL>/api/web/send?apikey= XXXXXXXXXXXXXXXX
&sender=XXXXXX&to=9900xxxxxx&message=TEST&format=xml

The response in XML format will be as shown below

```
<xml>
<groupID>62</groupID>
<MessageIDs>62-1</MessageIDs>
<status>AWAITED-DLR</status>
</xml>
```

Error Codes

When certain parameters of the HTTP API URL are not entered correctly, you will receive a response from the URL stating the type of error. These error codes will help you to pin-point the erroneous parameter in the URL and rectify it.

The common error codes received while using the HTTP API are tabulated below:

URL RESPONSE/ERROR CODE	DESCRIPTION
Invalid Input Data	One of the values are missing.
Account Inactive. Please contact Support	Your SMS account is inactive/deactivated.
Invalid Mobile Numbers	The mobile number entered is invalid/less that 10 Digits
Invalid Sender ID	The sender ID entered on the URL is either not applied for or has not been approved.
Insufficient Credits	Your SMS account is out of SMS credits.
Invalid Message GID	Invalid Message GID.
Invalid Schedule Date and Time!	Invalid Input is passed.
Invalid Schedule Date!	Date is not valid.
Invalid Schedule minutes, only 00, 15, 30, 45 allowed!	Minutes will only be accepted in multiples of 15.
Schedule time already past!	Time specified has already passed.

Sending an SMS

HTTP API sends a text message using POST/GET and form-encoded data. The URL you use to send a text message has a required field called working key, which is unique and assigned to your account.

The three required parameters for the Request are sender, to and message, which are used to define the destination phone number and the message body. If you want to send to multiple phone numbers, provide the numbers in comma separated format.

To send an SMS using the HTTP API, the following URL format is used:

http://<SMS_Service_URL>/api/web/send?apikey=<API_KEY>&to=<MOBILENUMBER>&sender=<SENDERID>&message=<MESSAGE>

FIELD NAME	DESCRIPTION
<API_KEY>	API key generated from SMS account.
<MOBILENUMBER>	The destination mobile number can be with or without +91 and 91.
<SENDERID>	Approved Sender ID
<MESSAGE>	The text message that needs to be sent. The message should be URL encoded.

For example;

http://<SMS_Service_URL>/api/web/send?apikey=<API_KEY>&to=9900XXXXXX&sender=DEMO&message=THISIS A TEST SMS

The URL shown above will send an SMS from your account to the mobile number 9900XXXXXX with the Sender ID as DEMO and the text "THIS IS A TEST SMS".

Method:

POST / GET

Request:

apikey=XXXXXXXXXXXXXXXXX&to=9900XXXXXX&sender=XXXXXX&message=hello

Response Formats:

The HTTP API response in the various supported formats are given below.

Text Response:

Message GID=360 ID=360-1

Invalid/DND Numbers:1

JSON Response:

{"groupID":61,"MessageIDs":"61-1","status":"AWAITED-DLR"}

XML Response:

```
<xml>
<groupID>62</groupID>
<MessageIDs>62-1</MessageIDs>
<status>AWAITED-DLR</status>
</xml>
```

HTTP URL Response

The responses you receive from the HTTP URL are tabulated below:

RESPONSE	DESCRIPTION
Invalid Input Data	One of the values are missing.
Invalid Mobile Numbers	The mobile number entered is invalid/less that 10 digits
Account Inactive. Please contact Support	Your SMS account is inactive/deactivated.
Invalid Sender ID	The sender ID entered on the URL is either not applied for or has not been approved.
Insufficient Credits	Your SMS account is out of SMS credits
Message GID=<GROUPID> ID	The message has been successfully submitted fordelivery. Displays the Message ID for further requests.

Note: The SMS you send can contain a maximum of 1000 characters. The creditsfor each SMS will be charged according to the length of each SMS.

Checking SMS Delivery Status

You can also view the delivery status of the SMS sent from your SMS account using the HTTP API. These delivery reports or statuses can be obtained for both, single and group messages.

Single Message Report (Pull)

The URL for retrieving a single SMS delivery report is in the following format:
http://<SMS_Service_URL>/api/status/message?apikey=<API_KEY>&messageid=123XXX

FIELD NAME	DESCRIPTION
<API_KEY>	API key generated from SMS account.
<MESSAGEID>	The Message ID that was previously display when submittingSMS.

For example;

http://<SMS_Service_URL>/api/status/message?apikey=<API_KEY>&messageid=123XXX

The URL shown above will display the delivery status of the SMS with the corresponding Message ID. The common statuses displayed are SUBMITTED, DELIVRD, UNDELIVRD, REJECTED etc.

Response Formats:

The HTTP API response in the various supported formats are given below.

JSON Response:

{"id":"363-1","Recipient":"+919900xxxxxx","status":"AWAITED-DLR"}

XML Response:

<xml>
<id>36-1</id>
<Recipient>+919900xxxxxx</Recipient><
status>AWAITED-DLR</status>
</xml>

Delivery Report URL Response

RESPONSE	DESCRIPTION
Invalid Input Data	One of the values are missing
Account Inactive. Please Contact Support	Your SMS account is inactive/deactivated.
Invalid Message ID	The Message ID does not exist
<STATUS>	Display the status of the SMS such as SUBMITTED, DELIVRD, UNDELIVRD, REJECTED

Obtaining Group Delivery Report (Pull)

The URL for retrieving a group SMS delivery report is in the following format:

http://<SMS_Service_URL>/status/group?apikey=<API_KEY>&messagegid=<MESSAGEGI D>

FIELD NAME	DESCRIPTION
<API_KEY>	API key generated from SMS account
<MESSAGEGID>	The message Group ID previously displayed when submitting SMS.

For example;

http://<SMS_Service_URL>/status/group?apikey=<API_KEY>&messagegid=123XXX

The URL shown above will retrieve the group SMS status of the corresponding Message IDs. The common responses are MESSAGEGID, DESTINATION NUMBER, SUBMITTED

Response Formats:

The HTTP API responses in the various supported formats are given below.

JSON Response:

```
[{"ID":"363-1","Recipient":"+919900xxxxxx","StatusDetails":"AWAITED-
DLR","UpdatedTime":"2013-06-17 14;40:59"}]
```

XML Response:

```
<xml>
        <sms><ID>363-1</ID>
        <Recipient>+919900xxxxxx</Recipient>
        <StatusDetails>AWAITED-DLR</StatusDetails>
        <UpdatedTime>2013-06-17 14;40:59</UpdatedTime>
        </sms>
</xml>
```

Delivery Report URL Response

RESPONSE	DESCRIPTION
Invalid Input Data	One of the values are missing
Account Inactive. Please Contact Support	Your SMS account is inactive/deactivated.
Invalid Message GID	Invalid Message GID
<STATUS>	Display the status of the SMS such as SUBMITTED, DELIVRD, UNDELIVRD, REJECTED

Checking Credit Availability in Your SMS Account

The URL for checking the credits available in your SMS account is in the following format: http://<SMS_Service_URL>/api/status/credit?apikey=<API_KEY>

FIELD NAME	DESCRIPTION
<API_KEY>	API key generated from SMS account.

For example;

http://<SMS_Service_URL>/api/status/credit?apikey=<API_KEY>

The preceding URL will fetch and display the number of credits available in your SMS account.

Credit Availability URL Response

RESPONSE	DESCRIPTION
Invalid Input Data	The value on the URL is missing
Account Inactive. Please Contact Support	Your SMS account is inactive/deactivated.
Your available credits is <VALUE>	Display the available credits in your account

Response Formats:

The HTTP API responses in the various supported formats are given below.

JSON Response:

```
{"Credits":"28028585"}
```

XML Response:

```
<xml>
        <Credits>28028585</Credits>
</xml>
```

HTTP URL Response

RESPONSE	DESCRIPTION
Invalid Input Data	One of the values are missing
Account Inactive. Please Contact Support	Your SMS account is inactive/deactivated.
INVALID NUMBERS : <MOBILENUMBER>,..	The mobile number entered is invalid/less than 10 digits.
Invalid Sender ID	The Sender ID type on the url is either no applied for or has not been approved
Insufficient Credits	Your SMS account is out of sms credits
Invalid Schedule Date and Time!	Invalid Input is passed
Invalid Schedule Date!	Date is not valid
Invalid Schedule minutes, only 00, 15, 30, 45 allowed!	Minutes will only accept the mentioned values

RESPONSE	DESCRIPTION
Schedule time already past!	Time specified has already passed
Message GID=<GROUPID> ID=<MESSAGEID>,<MESSAGEID2>	The message has been successfully submitted for delivery. Displays the Message ID for further request

Sending a Unicode SMS

The URL for sending a Unicode SMS is in the following format: http://<SMS_Service_URL>/api/web/send/?apikey=<API_KEY>&to=<MOBILENUMBER>&sender=<SENDERID>&message=<MESSAGE>&unicode=1

FIELD NAME	DESCRIPTION
<API_KEY>	API key generated from sms account.
<MOBILENUMBER>	The destination mobile number can be with or without +91 and 91.
<SENDERID>	Approved Sender ID
<MESSAGE>	The text message that needs to be sent. The message text should be in language you need to sendsms.
UNICODE=<1>	To send a Unicode SMS, the status should be 1

For example;
http://<SMS_Service_URL>/api/web/send?apikey=<API_KEY>&to=9900XXXXXX&sender=DEMO&message=xxxxxxxx&unicode=1

The preceding URL will send a Unicode SMS from your SMS account to the mobile number 9900XXXXXX with the sender ID as DEMO with the message textxxxxxxxxx.

Response Formats:

The HTTP API responses in the supported formats are given below.

Text Response:

Message GID=258 ID=258-1 Invalid/DND Numbers:

JSON Response:

{"groupID":384,"messages":[{"id":"258-1","mobile":"9900xxxxxx","status":"AWAITED_DLR"}]}

XML Response:

<xml>
<groupID>258</groupID>
<messages><sms><id>258-1</id>
<mobile>9900xxxxxx</mobile>
<status>AWAITED_DLR</status></sms></messages>
</xml>