

End User Manual  
For  
**HTTP(S)/XML/JSON API**

## Overview

HTTP(S)/XML/JSON API module is intended for users, who wants to use API SMS Gateway for sending messages. You can send single/multiple SMS using HTTP/HTTPS API format and XML/JSON API format. User name and Password for sending messages, will be provided by our Team after user opens his account.

## System Overview

### HTTP/HTTPS API

HTTP/HTTPS API supports requirements given below.

1. Supports all languages with special characters
2. Supports concatenated messages (long SMS)
3. Supports text and Unicode messages (Hindi, Gujarati, Marathi etc)
4. Dynamic sender identity

### URL:

You will call URL given below, using GET method.

<http://49.50.67.32/smsapi/httpapi.jsp?>

OR

<https://49.50.67.32/smsapi/httpapi.jsp?>

Please mention the below parameters for above URL request:

### Compulsory Parameter Description

1. username: Valid User Name provided, when user registers for an account.
2. password: Valid Password provided in combination with the Username, when user registers for an account.
3. to: comma separated mobile numbers on which message will be delivered. (Example 98XXXXXXXXX).
4. from: from parameter refers to the Sender Id registered for sending messages
5. text: text parameter refers to the Message content, which needs to be sent to entered Mobile Number. (Example: Written content will be sent.)

## Optional Parameter Description

1. coding: If you want to send Unicode SMS, then enter value as 2. If you do not use this parameter, then it will take non-Unicode by default.
2. scheduletime: Schedule time to deliver this message in the format of YYYY-MM-DD HH:mm
3. flash: If you want to send Flash SMS, then enter value as per below requirement:
  - ✓ 1 : This type supports all modern handsets and English SMS count will be as 70 chars per SMS (like non –English SMS).
  - ✓ 2 : This type might be not supports all modern handsets and English SMS count will be as 160 chars per SMS (like other English SMS).
4. responsetype: you can get response in text by using 'csv' and in JSON by using 'json' as value in this parameter. Default response type will be XML if you are not using this parameter.

## Example to call HTTP(S)API:

<http://49.50.67.32/smsapi/httpapi.jsp?username=XXXXXXXXXX&password=XXXXXX&from=XXXXXX&to=XXXXXXXXXX,XXXXXXXXXX&text=hello+world&coding=0&scheduletime=XXXX-XX-XX XX:XX&flash=1>

OR

<https://49.50.67.32/smsapi/httpapi.jsp?username=XXXXXXXXXX&password=XXXXXX&from=XXXXXX&to=XXXXXXXXXX,XXXXXXXXXX&text=hello+world&coding=0&scheduletime=XXXX-XX-XX XX:XX&flash=2>

## Response:

You will receive below XML response if Message is Successfully Sent.

```
<data>
<ackid>xxxxx-xxxxxxxxx </ackid>
<msgid>xxxxxx-xxxxxxxxxxxx, xxxxxxxxxxx</msgid>
</data>
```

You will receive below XML response if Message is not successfully sent.

```
<Error>
<ErrorCode>xxx</ErrorCode>
<ErrorDesc>xxxxxxxxxxxxxxxx</ErrorDesc>
</Error>
```

## XML API

For bulk SMS sending, write XML data in format given below and post to URL given.

### XML Format for Single SMS to Multiple Numbers:

URL: You will call the below URL using POST method.

<http://49.50.67.32/mobxmlapi/xmlsmsapi.jsp>

OR

<https://49.50.67.32/mobxmlapi/xmlsmsapi.jsp>

```
<xml>
<data>
<userinfo>
<name>XXXXXXXX</name>
<password>XXXXXXXX</password>
<from>XXXXXX</from>
</userinfo>
<recipient>
<to>XXXXXXXXXX</to>
</recipient>
<recipient>
<to>XXXXXXXXXX</to>
</recipient>
<recipient>
<to>XXXXXXXXXX</to>
</recipient>
<recipient>
<to>XXXXXXXXXX</to>
.
.
.
.
.
.
.
.
</recipient>
<message>
<text>This is testing for xml data</text>
<coding>0</coding>
</message>
</data>
```



```
<recipient>
<to>XXXXXXXXXX</to>
<msg>This is testing for xml data</msg>
</recipient>
<recipient>
<to>XXXXXXXXXX</to>
<msg>This is testing for xml data</msg>
</recipient>
</data>
```

### **XML Format for Single SMS to Multiple Numbers & Multiple SMS to Multiple Numbers:**

URL: You will call the below URL using POST method.

<http://49.50.67.32/nxmlapi/xmlsmsapi.jsp>

OR

<https://49.50.67.32/nxmlapi/xmlsmsapi.jsp>

```
<xml>
<data>
<user>
<name>XXXXXXXX</name>
<password>XXXXXXXX</password>
<from>XXXXXX</from>
<to>XXXXXXXX</to>
<text> This is testing for xml data </text>
<coding>0</coding>
</user>
<user>
<name>XXXXXXXX</name>
<password>XXXXXXXX</password>
<from>XXXXXX</from>
<to>XXXXXXXX</to>
<text> This is testing for xml data </text>
<coding>0</coding>
</user>
<user>
<name>XXXXXXXX</name>
<password>XXXXXXXX</password>
<from>XXXXXX</from>
<to>XXXXXXXX</to>
<text> This is testing for xml data </text>
<coding>0</coding>
</user>
<user>
.
.
.
.
</user>
</data>
```

## Compulsory Parameter Description

1. username: Valid User Name provided, when user registers for an account
2. password: Valid Password provided in combination with the Username, when user registers for an account.
3. to: comma separated mobile numbers on which message will be delivered. (Example 98XXXXXXXX).
4. from: from parameter refers to the Sender Id registered for sending messages
5. text: text parameter refers to the Message content, which needs to be sent to entered Mobile Number. (Example: Written content will be sent.)

## Optional Parameter Description

1. coding: If you want to send Unicode SMS, then enter value as 2. If you do not use this parameter, then it will take non-Unicode by default.

## Response:

You will receive below XML response if Message is successfully sent.

```
<data>
<ackid>xxxxx-xxxxxxxxxx </ackid>
<msgid>xxxxxx-xxxxxxxxxxxx, xxxxxxxxxxx</msgid>
</data>
```

You will receive below XML response if Message is not successfully sent.

```
<Error>
<ErrorCode>xxx</ErrorCode>
<ErrorDesc>xxxxxxxxxxxxxxxx</ErrorDesc>
</Error>
```

## JSON API

For bulk SMS sending, write JSON data in format given below and post to URL given.

### JSON Format for Single SMS to Single/Multiple Numbers:

URL: You will call the below URL using POST method.

<http://49.50.67.32/smsapi/jsonapi.jsp>

OR

<https://49.50.67.32/smsapi/jsonapi.jsp>

```
{
"username" : "XXXXXXXXXX",
"password" : "XXXXXXXXXX",
"from" : "XXXXXXX",
"to" : [ "XXXXXXXXXXXX", "XXXXXXXXXXXX", "XXXXXXXXXXXX", "XXXXXXXXXXXX", ..... "XXXXXXXXXXXX" ],
"text" : "This is testing for json data",
"coding" : "0",
"flash" : "1",
"scheduletime" : "XXXX-XX-XX XX:XX"
}
```

### Compulsory Parameter Description

1. username: Valid User Name provided, when user registers for an account
2. password: Valid Password provided in combination with the Username, when user registers for an account.
3. to: comma separated mobile numbers on which message will be delivered. (Example 98XXXXXXXXX).
4. from: from parameter refers to the Sender Id registered for sending messages
5. text: text parameter refers to the Message content, which needs to be sent to entered Mobile Number. (Example: Written content will be sent.)



## Optional Parameter Description

1. coding: If you want to send Unicode SMS, then enter value as 2. If you do not use this parameter, then it will take non-Unicode by default.
2. scheduletime: Schedule time to deliver this message in the format of YYYY-MM-DD HH:mm
3. flash: If you want to send Flash SMS, then enter value as per below requirement:
  - ✓ 1 : This type supports all modern handsets and English SMS count will be as 70 chars per SMS (like non –English SMS).
  - ✓ 2 : This type might be not supports all modern handsets and English SMS count will be as 160 chars per SMS (like other English SMS).

## Response:

You will receive below JSON response if Message is successfully sent.

```
{
  "data" : {
    "ackid" : "xxxxx-xxxxxxxxxx",
    "msgid" : [ "xxxxxx-xxxxxxxxxxxx, xxxxxxxx", "xxxxxx-xxxxxxxxxxxx, xxxxxxxx",.....]
  }
}
```

You will receive below JSON response if Message is not successfully sent.

```
{
  "Error" : {
    "ErrorCode" : "xxx",
    "ErrorDesc" : "xxxxxxxxxxxxxxxxxx"
  }
}
```

## Pull DLR API

URL for getting delivery report:

<http://49.50.67.32/dlrresponse/dlrresp.jsp?ackid=xxxxxx>

OR

<https://49.50.67.32/dlrresponse/dlrresp.jsp?ackid=xxxxxx>

### Parameter Description:

1. ackid: ackid is the parameter, for which you want DLR and this should be same as you have got in response for requested SMS through HTTP/XML API.
2. responsetype: you can get response in text by using 'csv' and in JSON by using 'json' as value in this parameter. Default response type will be XML if you are not using this parameter.

### Response:

You will receive below XML response.

```
<dlr>
<dlrinfo>
<msgid>xxxxx-xxxxxxxxxxxx</msgid>
<ackid>xxxx-xxxxxxxxxxxx</ackid>
<mob>91xxxxxxxx</mob>
<senderid>xxxxx</senderid>
<submittime>xxxx-xx-xx xx:xx:xx.x</submittime>
<delvtime> xxxx-xx-xx xx:xx:xx.x </delvtime>
<dlrdesc>xxxxxxxxxxxx</dlrdesc>
</dlrinfo>
<dlrinfo>
<msgid> xxxxx-xxxxxxxxxxxx </msgid>
<ackid> xxxxxxxxxxxxxxxxx </ackid>
<mob>91xxxxxxxx</mob>
<senderid>xxxxx</senderid>
<submittime> xxxx-xx-xx xx:xx:xx.x </submittime>
<delvtime> xxxx-xx-xx xx:xx:xx.x </delvtime>
<dlrdesc>xxxxxxxxxxxx </dlrdesc>
</dlrinfo>
</dlr>
```

## Push DLR

Client's URL need to be configured at our end in order to push the DLR with below Parameters:

<b>message_id</b>	(For Exp: username-1484380575027)
<b>dlrstatus</b>	(For Exp: Delivered)
<b>msisdn</b>	(For Exp: 9266669268)
<b>sender_id</b>	(For Exp: TESTIN)
<b>submittime</b>	(For Exp: 2017-04-14 13:26:00)
<b>delvtime</b>	(For Exp: 2017-04-14 13:27:00)
<b>dlrcode</b>	(For Exp: 000)

Client will receive above parameters on their configured URL.

## **Error Codes with Description**

<b>Code</b>	<b>Description</b>
010	Sender Id length/format error
011	MSISDN Length/Format Error
15	Insufficient Credit Error
103	Message Empty Error
104	Coding Error
105	Scheduletime Format Error

\*\*NOTE: MESSAGE HAS TO BE URL ENCODED FORMAT

\*\*NOTE: MULTIPLE MOBILE NUMBERS HAS TO BE COMMA SEPERATED

**For any support, please contact our Team**

\*\*\*\*\*End of Document\*\*\*\*\*