HTTPS/XML API Hand Book

Table of Contents

List Of Tables

Preface

The API user guide provides the specifications of the HTTPS based API's developed at NIC to send automated bulk sms via the internet. This messaging service uses the NIC gateway to send bulk SMSs. The NIC gateway not only sends bulk SMS but also sends delivery reports to its customers, based on the requirements.

Audience

This guide is intended for developers who plan to integrate their system with NIC SMS service.

Assumptions

The guide assumes that you are familiar with and have sufficient experience in applying the following technologies and concepts of HTTP and XML.

Abbreviations and Acronyms used in the guide.

QS	Query String
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
ACK	Acknowledgement
XML	Extensible Markup Language
HTT	Hyper Text Transfer
Protocol ACK	Acknowledgement

Table 1. Abbreviations and Acronyms.

SMS Gateway HTTPS Query String API

SMS Gateway provides HTTPS Uniform Resource Location (URL) to enterprises to integrate messaging services within their applications. On any event occurrence within the enterprise application, a simple HTTP request can be generated with this URL to send a message to enterprise end users. Query String interface takes care of receiving client requests via Query String over HTTPS protocol.

HTTPS Interface

Query String interface takes care of receiving client requests via Query String over HTTPS protocol.

- HTTPS URL

Following is the URL for the Query string interface,

https://IP:PORT/failsafe/HTTPLink?

The IP and PORT will be provided at the time of account creation.

- HTTPS PARAMETERS

The below mentioned are the HTTPS Parameters,

S.No	Parameter	Type	Description
1	username *	String	Account id; will be provided by SMS Platform
2	pin *	String	Secret password; will be provided by SMS Platform
3	Mnumber *	String	Mobile number with country code
4	message *	String	SMS Text content
5	Sche Time	String	Scheduled time to deliver this message in the format of yyyy/MM/dd/HH/mm; default is null
6	Signature**	String	Sender id / from address; This will be configured by SMS Platform during account creation.
7	expiry	Number	In minutes:(example: 20 represents the message should expire if not delivered to SMSC within 20 minutes from the message accepted time of SMS Platform). Value 0 represents no expiry and default value is 0.

8	splitAlgm	String	split – message will be splits based on operator restriction (140/160 for English) and delivered; trunc – message will be truncated after the split based on operator restriction; concat – message will be concatenated while delivery but billing will happen based on number of split messages.
9	Custref	String	Client generated unique id for client's reference; It should be numeric and less than 30 digits.
10	Billref	String	Client generated bill reference id; It should be numeric and less than 6 digits.
11	msgType *** (it is not required in MLink)	String	PM – Plain text message; UC – Unicode Message; BM – Binary text message(ringtone, logo, picture, wap link); FL – Flash message; SP – messages to special port; default is PM
12	Prty	Number	1 – High; 0 – Low; default is 0
13	Udh	String	User data header info for binary messages;
14	Port	Number	Port to deliver the message. Max Length – 6
15	Dlrtype	Number	0 – No need for dlr; 1 – end delivery notification success or failure 2 – end delivery notification failure only 4 – SMS Platform failures / reject status only; 5 - SMS Platform failures / reject status + end delivery notification success or failure; 6 - SMS Platform failures/reject status + end delivery notification failure; default is 0
16	dlt_entity_id	String	xxxxxxxx
17	dlt_template_id	String	xxxxxxxx

TABLE 2: HTTPS PARAMETERS

Note :

- *In the upper parameters (*) represents mandatory*
- *msgType parameter (***) is not required in MLink*

RESPONSE

Query String Interface response contains the platform's acknowledgement id, API success/error code, success/error description and platform server's request accepted timestamp.

Following is the response format for HTTPS Query string interface,

Message Accepted for Request Id=xxxxxxxxxxxxxx~code=APIcode & info=status description & Time=Timestamp

Here,

- 'xxxxxxxxxxx' is the platform's acknowledgement id.
- 'API CODE' is the platforms success or error code.
- 'STATUS DESCRIPTION' is the platforms success or error description.
- 'TIME STAMP' is the platforms server request accepted date and time.

The following table contains the possible API codes and description in the response,

S. No.	Error Code	Description
1	-2	Invalid credentials
2	-3	Empty mobile number
3	-4	Empty message
4	-5	HTTPS disabled
5	-6	HTTP disabled
6	-410	Invalid Destination Address
7	-201	Email Delivery Disabled
8	-404	Invalid MsgType
9	-406	Invalid Port
10	-407	Invalid Expiry minutes
11	-408	Invalid Customer Reference Id
12	- 433	Invalid Customer Reference Id Length
13	-409	Invalid Bill Reference Id

14	-432	Invalid Bill Reference Id Length
15	-13	Internal Error
16	-401	Invalid Scheduled Time
17	000	SMS Platform Accepted

TABLE 3. API CODES & DESCRIPTIONS.

Example

- Sample Url Format

https://smsgw.sms.gov.in/failsafe/HttpLink?username=xxxxxx&pin=xxxxxxx&message=message&mnumber = 91XXXXXXXXXX&signature= SENDER ID&dlt_entity_id=xxxxxxx&dlt_template_id=xxxxxxx

- Sample Success Response

Message Accepted For Request ID=123145785964125468795 4~code=API000 & info=Sms platform accepted & Time = 2007/10/04/09/58

- Sample Error Response

Message Rejected for Request ID= ~API-2 & info=REJECTED: Invalid credentials & Time = 2007/10/04/10/31

Sample URLs For Message Types

- Plain Message

https://smsgw.sms.gov.in/failsafe/HttpLink?username=xxxxxx&pin=xxxxxx&message=textmessage&mnumber= 91XXXXXXXXXX&signature=SENDER ID&dlt_entity_id=xxxxxxx&dlt_template_id=xxxxxxx

- Multilingual Message
    - Unicode Message

https://smsgw.sms.gov.in/failsafe/HttpLink?username=xxxx&pin=xxxxx&message=0915094D092F093E0020 0917093E09020917094109320940002009150 94B&mnumber=91XXXXXXXXXX&signature= SENDER ID&dlt_entity_id=xxxxxxx&dlt_template_id=xxxxxxx

- MLink Message

https://smsgw.sms.gov.in/failsafe/MLink?username=xxxx&pin=xxxx&message=हिन्दी%20বাংলা%20मराठी%20తెలుగు%20தமிழ்%20ગુજરાતી&mnumber=91XXXXXXXXXX&signature=SENDERID&dlt_entity_id=xxxxxxx&dlt_template_id=xxxxxxx

Note : msgType parameter is not required in MLink

SMS Gateway XML Interface

The XML API Reference Guide will help you use SMS Gateway India's platform to send SMS messages through XML over HTTP. This document provides the necessary information relating to the XML Interface.

XML Interface

XML interface is taking care of receiving client requests via XML over HTTPS protocol. Supports only POST method. Three kinds of XML format supported by SMS PLATFORM,

- Single message to single recipient
- Single message to multiple recipients
- Multiple Message to multiple recipients

Single Message to Single Recipient

URL - http://smsgw.sms.gov.in/failsafe/HttpData_SM

Request XML

Following is the format of request XML for sending single message to single recipient

```
<? xml version="1.0”?>
<a2wml version="2.0" >
<request username="rajesh21" pin="rajesh123" >
<from Address>Air2web</from Address>
<recipient List>
  <destAddress>919884360784</destAddress>
</recipient List>
<message>
 <message Type>PM</message Type>
<port></port>
  <udh></udh>
  <messageTxt>type your message here</messageTxt>
  <odRequestId></odRequestId >
  <custref></custref >
  <billref></billref>
  <splitAlgm></splitAlgm>
  <scheduleTime></scheduleTime>
  <expiryMinutes></expiryMinutes>
  <dlrtype>0</dlrtype>
  <language Code></language Code>
  <priority></priority>
  <country></country>
<dlt_template_id>123</dlt_template_id>
<dlt_entity_id>456</dlt_entity_id>
</message>
</request>
</a2wml>
```

Response XML

```
<? xml version="1.0"?>
<a2wml>
<response>
<version>2.0</version>
<timestamp>2009/03/03/04/58</timestamp>
<request ID>89112360797316231002997</request ID>
<code>API000</code>
<info>Platform accepted</info>
</response>
</a2wml>
```

Single Message To Multiple Recipients

URL - https://smsgw.sms.gov.in/failsafe/HttpData_SM

Requested Xml

Following is the format of Request of XML for sending single Message to Multiple Recipients

```
<?xml version="1.0" ?>
<a2wml version="2.0" >
<request username="rajesh21" pin="rajesh123" >
<from Address></from Address>
<recipient List>
  <recipient>
     <destAddress>9884360784</destAddress>
     <custref></custref>
     <billref></billref>
     <dlrtype></dlrtype>
     <priority></priority>
     <scheduleTime></scheduleTime>
     <expiryMinutes></expiryMinutes>
     <country></country>
  </recipient>
  <recipient>
     <destAddress>9884360785</destAddress>
     <custref></custref>
     <billref></billref>
     <dlrtype></dlrtype>
     <priority></priority>
     <scheduleTime></scheduleTime>
     <expiryMinutes></expiryMinutes>
     <country></country>
  </recipient>
  <recipient>
```

```
      <destAddress>9884360786</destAddress>
      <custref></custref>
      <billref></billref>
      <dlrtype></dlrtype>
      <priority></priority>
      <scheduleTime></scheduleTime>
      <expiryMinutes></expiryMinutes>
      <country></country>
   </recipient>
</recipient List>
<message>
   <message Type></message Type>
   <port></port>
   <udh></udh>
   <messageTxt>type your message here</messageTxt>
   <splitAlgm></splitAlgm>
   <language Code></language Code>
     <dlt_template_id>123</dlt_template_id>
     <dlt_entity_id>456</dlt_entity_id>
</message>
</request>
</a2wml>
```

Response XML

```
<?xml version="1.0"?>
<a2wml>
<response>
<version>2.0</version>
<timestamp>2009/03/03/05/03</timestamp>
<request ID>89112360800366201005994</request ID>
<code>API000</code>
<info>Platform accepted</info>
</response>
</a2wml>
```

Multiple Messages to Multiple Recipients

URL - https://smsgw.sms.gov.in/failsafe/HttpData_MM

Request XML

Following is the format of Request XML for Sending Multiple Messages to Multiple Recipients.

```
<?xml version="1.0" ?>
<a2wml version="2.0" >
<request username="rajesh21" pin="rajesh123"  >
<messageList>
<fromAddress>Air2web</fromAddress>
  <destAddress>919884360784</destAddress>
  <messageType>PM</messageType>
  <port></port>
  <udh></udh>
  <messageTxt>XML mm</messageTxt>
  <odRequestId></odRequestId>
  <custref></custref>
  <billref></billref>
  <splitAlgm></splitAlgm>
  <scheduleTime></scheduleTime>
  <expiryMinutes></expiryMinutes>
  <dlrtype></dlrtype>
  <languageCode></languageCode>
  <priority></priority>
  <country></country>
  <dlt_template_id>123</dlt_template_id>
  <dlt_entity_id>456</dlt_entity_id>
</messageList>
<messageList>
<fromAddress>Air2web</fromAddress>
  <destAddress>919884360785</destAddress>
  <messageType>PM</messageType>
  <port></port>
  <udh></udh>
  <messageTxt>XML mm</messageTxt>
  <odRequestId></odRequestId>
  <custref></custref>
  <billref></billref>
```

```
  <scheduleTime></scheduleTime>
  <expiryMinutes></expiryMinutes>
  <dlrtype></dlrtype>
  <languageCode></languageCode>
  <priority></priority>
  <country></country>
  <dlt_template_id>123</dlt_template_id>
  <dlt_entity_id>456</dlt_entity_id>

</messageList>
<messageList>

<fromAddress>Air2web</fromAddress>
  <destAddress>919884360786</destAddress>
  <messageType>PM</messageType>
  <port></port>
  <udh></udh>
  <messageTxt>XML mm</messageTxt>
  <odRequestId></odRequestId>
  <custref></custref>
  <billref></billref>
 <splitAlgm></splitAlgm>
  <scheduleTime></scheduleTime>
  <expiryMinutes></expiryMinutes>
  <dlrtype></dlrtype>
  <languageCode></languageCode>
  <priority></priority>
  <country></country>
  <dlt_template_id>123</dlt_template_id>
  <dlt_entity_id>456</dlt_entity_id>
</messageList>

<messageList>
 <fromAddress>Air2web</fromAddress>
  <destAddress>919884360787</destAddress>
  <messageType>PM</messageType>
 <port></port>
  <udh></udh>
  <messageTxt>XML mm</messageTxt>
  <odRequestId></odRequestId>
  <custref></custref>
  <billref></billref>
```

```
      <splitAlgm></splitAlgm>
      <scheduleTime></scheduleTime>
      <expiryMinutes></expiryMinutes>
      <dlrtype></dlrtype>
      <languageCode></languageCode>
      <priority></priority>
      <country></country>
      <dlt_template_id>123</dlt_template_id>
      <dlt_entity_id>456</dlt_entity_id>
 </messageList>

 <messageList>
 <fromAddress>Air2web</fromAddress>
   <destAddress>919884360788</destAddress>
   <message Type>PM</message Type>
 <port></port>
   <udh></udh>
   <messageTxt>XML mm</messageTxt>
   <odRequestId></odRequestId>
   <custref></custref>
   <billref></billref>
   <splitAlgm></splitAlgm>
   <scheduleTime></scheduleTime>
   <expiryMinutes></expiryMinutes>
   <dlrtype></dlrtype>
   <language Code></language Code>
   <priority></priority>
<country></country>
   <dlt_template_id>123</dlt_template_id>
   <dlt_entity_id>456</dlt_entity_id>

 </message List>
 </request>
 </a2wml>
```

Response XML

```
<?xml version="1.0"?>
<a2wml>
<response>
<version>2.0</version>
<timestamp>2009/03/03/05/03</timestamp>
<request ID>8911236080036620100594</request ID>
<code>API000</code>
<info>Platform accepted</info>
</response>
</a2wml>
```

Request XML Nodes and Attributes Values

S.No	Parameter	Type	Description
1	username *	String	Account id; will be provided by SMS Platform
2	pin *	String	Secret password; will be provided by SMS Platform
3	mnumber *	String	Mobile number with country code
4	message *	String	SMS Text content
5	scheTime	String	Scheduled time to deliver this message in the format of yyyy/MM/dd/HH/mm; default is null
6	Signature**	String	Sender id / from address; This will be configured by SMS Platform during account creation.
7	Expiry	Number	In minutes; example: 20 represents the message should expire if not delivered to SMSC within 20 minutes from the message accepted time of SMS Platform. Value 0 represents no expiry; default
8	splitAlgm	String	split – message will be splits based on operator restriction (140/160 for English) and delivered; trunc – message will be truncated after the split based on operator restriction; concat – message will be concatenated while delivery but billing will happen based on number of split messages.
9	Custref	String	Client generated unique id for client's reference; It should be numeric and less than 30 digits.
10	Billref	String	Client generated bill reference id; It should be numeric and less than 6 digits.
11	msgType	String	PM – Plain text message; UC – Unicode Message; BM – Binary text message(ringtone, logo, picture, wap link); FL – Flash message; SP – messages to special port; default is PM
12	Prty	Number	1 – High; 0 – Low; default is 0
13	Udh	String	User data header info for binary messages;

14	Port	Number	Port to deliver the message. Max Length – 6
15	Dlrtype	Number	0 – No need for dlr; 1 – end delivery notification success or failure 2 – end delivery notification failure only 4 – SMS Platform failures / reject status only; 5 - SMS Platform failures / reject status + end delivery notification success or failure; 6 - SMS Platform failures/reject status + end delivery notification failure; default is 0

TABLE 4: Request XML Nodes & Attributes Values.

NOTE: In the above table (*) represented fields are mandatory.

Response XML Nodes and Attributes Values

S.No	Node/Attribute	Type	Description
1	Version	Number	Release version of SMS platform
2	Timestamp	String	SMS platform accepted time stamp in the format of yyyy/MM/dd/HH24/mm
3	Request Id	Number	SMS platform message acknowledgement id of length 23
4	Code	String	Platform API code
5	Info	String	Description of Platform API code

TABLE 5: Response XML Nodes & Attributes Values.

API Codes and Description in their Response.

S. No.	Error Code	Description
1	-7	Invalid credentials
2	-13	Internal Error
3	-8	Invalid XML
4	000	SMS Platform Accepted

TABLE 6: API Codes & Description

International Messages

SMS platform also supports international messaging. Below are the list of supported countries.

Country	countrycode
UNITED STATES OF AMERICA	1
RUSSIAN FEDERATION	7
EGYPT	20
SOUTH AFRICA	27
GREECE	30
NETHERLANDS	31
BELGIUM	32
FRANCE	33
SPAIN	34
HUNGARY	36
ITALY-VATICAN	39
ROMANIA	40
SWITZERLAND	41
AUSTRIA	43
UK	44
DENMARK	45
SWEDEN	46
NORWAY	47
POLAND	48
GERMANY	49
PERU	51

MEXICO	52
CUBA	53
ARGENTINE	54
BRAZIL	55
CHILE	56
COLOMBIA	57
VENEZUELA	58
MALAYSIA	60
MALAYSIA	60
MALAYSIA	60
AUSTRALIA	61
AUSTRALIA	61
INDONESIA	62
PHILIPPINES	63
NEW ZEALAND	64
SINGAPORE	65
THAILAND	66
KAZAKHSTAN	73
JAPAN	81
KOREA	82
VIETNAM	84
CHINA	86
TURKEY	90
INDIA	91
PAKISTAN	92
AFGHANISTAN	93
SRI LANKA	94
MYANMAR	95
IRAN	98
MOROCCO	212
ALGERIA	213
TUNISIA	216
LIBYA	218
GAMBIA	220
SENEGAL	221
MAURITANIA	222
MALI	223
GUINEA	224

CTTE D'IVOIRE	225
BURKINA FASO	226
NIGER	227
TOGOLESE REPUBLIC	228
BENIN	229
MAURITIUS	230
LIBERIA	231
SIERRA LEONE	232
GHANA	233
GHANA	233
GHANA	233
NIGERIA	234
CHAD	235
CENTRAL AFRICAN REPUBLIC	236
CAMEROON	237
CAPE VERDE	238
EQUATORIAL GUINEA	240
GABONESE	241
CONGO	242
CONGO REPUBLIC	243
ANGOLA	244
SEYCHELLES	248
SUDAN	249
RWANDESE REPUBLIC	250
ETHIOPIA	251
SOMALI	252
DJIBOUTI	253
KENYA	254
KENYA	254
KENYA	254
KENYA	254
KENYA	254
TANZANIA	255
UGANDA	256
BURUNDI	257
MOZAMBIQUE	258
ZAMBIA	260
MADAGASCAR	261

REUNION	262
ZIMBABWE	263
NAMIBIA	264
MALAWI	265
LESOTHO	266
BOTSWANA	267
SWAZILAND	268
COMOROS	269
ERITREA	291
ARUBA	297
FAROE ISLANDS	298
GREENLAND	299
GIBRALTAR	350
PORTUGAL	351
LUXEMBOURG	352
IRELAND	353
ICELAND	354
ALBANIA	355
MALTA	356
CYPRUS	357
FINLAND	358
BULGARIA	359
LITHUANIA	370
LATVIA	371
ESTONIA	372
MOLDOVA	373
ARMENIA	374
BELARUS	375
ANDORRA	376
MONACO	377
SAN MARINO	378
UKRAINE	380
YUGOSLAVIA	381
MONTENEGRO	382
CROATIA	385
SLOVENIA	386
BOSNIA AND HERZEGOVINA	387
YUGOSLAV MACEDONIA	389

CZECH REPUBLIC	420
SLOVAK REPUBLIC	421
LIECHTENSTEIN	423
EL SALVADOR	503
NICARAGUA	505
COSTA RICA	506
PANAMA	507
HAITI	509
BOLIVIA	591
GUYANA	592
ECUADOR	593
FRENCH GUIANA	594
PARAGUAY	595
SURINAME	597
URUGUAY	598
NETHERLANDS ANTILLES	599
EAST TIMOR	670
BRUNEI DARUSSALAM	673
PAPUA NEW GUINEA	675
FIJI	679
COOK ISLANDS	682
NIUE	689
KOREA	850
HONGKONG	852
MACAU	853
CAMBODIA	855
LAOS	856
BANGLADESH	880
TAIWAN	886
MALDIVES	960
LEBANON	961
JORDAN	962
SYRIA	963
IRAQ	964
KUWAIT	965
SAUDI ARABIA	966
YEMEN	967
OMAN	968

PALESTINIAN TERRITORY	970
UNITED ARAB EMIRATES	971
ISRAEL	972
BAHRAIN	973
QATAR	974
BHUTAN	975
MONGOLIA	976
NEPAL	977
TAJIKISTAN	992
TURKMENISTAN	993
AZERBAIJANI REPUBLIC	994
GEORGIA	995
KYRGYZ REPUBLIC	996
UZBEKISTAN	998
CANADA	1204
CANADA	1226
CANADA	1236
CANADA	1249
CANADA	1250
CANADA	1289
CANADA	1306
CANADA	1343
Cayman Islands	1345
CANADA	1365
CANADA	1403
CANADA	1416
CANADA	1418
CANADA	1431
CANADA	1437
CANADA	1438
CANADA	1450
CANADA	1506
CANADA	1514
CANADA	1519
CANADA	1579
CANADA	1581
CANADA	1587
CANADA	1604

CANADA	1613
CANADA	1639
CANADA	1647
CANADA	1705
CANADA	1709
ST. LUCIA	1758
DOMINICA	1767
CANADA	1778
CANADA	1780
CANADA	1807
DOMINICAN REPUBLIC	1809
CANADA	1819
CANADA	1829
CANADA	1849
CANADA	1867
Trinidad and Tobago	1868
CANADA	1873
JAMAICA	1876
CANADA	1902
CANADA	1905

Sample Codes

ASP:

```
<%@ Language=VBScript %>
<%
'
' Example of using the SMS platform from ASP using VBScript.
'

baseURL =
"https://smsgw.sms.gov.in/failsafe/HttpLink?username=<USERNAME>&pin=<PASSWORD>
&"
replyTo = "<SENDERID>"
```

```
recipient = "<MOBILENO>"
messageBody = "Test message sent via SMS platform using VBScript"

' URL encode message body
messageBody = Server.URLEncode(messageBody)
' Construct URL
URI = URI & baseURL
URI = URI & "signature=" & replyTo
URI = URI & "&mnumber=" & recipient
URI = URI & "&message=" & messageBody
Set http = Server.CreateObject("Microsoft.XmlHttp")
http.open "GET", URI, False
http.send ""
Response.write(http.responseText)
Set http = Nothing
%>
```

JAVA:

```
//
// Example of using the SMS platform from Java.
//
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;

public class HTTPSampleCode {
        public static void main(String[] args) {
                try {
```

```
String baseURL =
"https://smsgw.sms.gov.in/failsafe/HttpLink?username=<USERNAME>&pin=<PASSWORD>
&";
String replyTo = "<SENDERID>";
String recipient = "<MOBILENO>";
String messageBody = "Test message sent via SMS platform using
Java";
// URL encode message body
messageBody = URLEncoder.encode(messageBody, "UTF-8");
// Construct URL
StringBuffer URI = new StringBuffer();
URI.append(baseURL);
URI.append("signature=" + replyTo);
URI.append("&mnumber=" + recipient);
URI.append("&message=" + messageBody);
String result = "";
// Open connection and send request
URL url = new URL(URI.toString());
URLConnection conn = url.openConnection();
// Get the response
BufferedReader rd = new BufferedReader(new InputStreamReader(
        conn.getInputStream()));
StringBuffer sb = new StringBuffer();
String line;
while ((line = rd.readLine()) != null) {
    sb.append(line);
}
rd.close();
// Print results
result = sb.toString();
System.out.println("Result:" + result);
```

```
                } catch (Exception e) {

                        e.printStackTrace();
                }
        }
}
```

PHP:

```
<?php
//
// Example of sending messages through the Air2Web DirectTEXT message delivery API with PHP.
//
if (function_exists("curl_init"))
{
        // initialize a new curl resource
        $ch = curl_init();

        $baseURL = "https://smsgw.sms.gov.in/failsafe/HttpLink?username=<USERNAME>&pin=<PASSWORD>&
";
        $replyTo = "<SENDERID>";
        $recipient = "<MOBILENO>";
        $messageBody = "Test message sent via SMS platform using PHP";
        // URL encode message body
        $messageBody = urlencode($messageBody);
        $URI = $baseURL;
        $URI .= "signature=" . $replyTo;
        $URI .= "&mnumber=" . $recipient;
```

```
        $URI .= "&message=" . $messageBody;
        // Set URL to connect to
        curl_setopt($ch, CURLOPT_URL, $URI);
        // Set header supression
        curl_setopt($ch, CURLOPT_HEADER, 0);
    // Disable SSL peer verification
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

        // Indicate that the message should be returned to a variable
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        // Make request
        $content = curl_exec($ch);
        //print responses
        print($content);
        // Clean up
        curl_close($ch);
}
else
{
  print("ERROR: curl library is not installed");
}
?>
```

WGET:

```
wget “https://secure-mrr.air2web.com/a2w preRouter/http/httpApiRouterreply
to=12345&recipient=14045552900&body=Test%20message%20%sent%20Air2web%20DirectTEXT
%
20from%20wget”
```

DLT changes required at SMPP packet

To Support DLT changes in NIC SMPP we have added two additional TLV parameters

- **Supports customized TLV parameters in Submit_sm.**

Tag Name	Tag Value
DLT Template ID	1401
DLT Entity ID	1400

We have checked with below configuration in our system and Below are the configuration for your reference.

====== This is for reference Only ==========
Sample for ESME using kannel with new DLT parameters

ESME - Kannel / Version 1.4.5 - stable
Operating System - Centos 7.6 / 64 bit

In kannel configuration need to add these two TLV for entity ID and template ID

group = smpp-tlv
name = entityid
tag = 0x1400
type = octetstring
length = 20
smsc-id = <smsc-id1>,<smsc-id2> ..

group = smpp-tlv
name = templateid
tag = 0x1401
type = octetstring
length = 20
smsc-id = <smsc-id1>,<smsc-id2> ..

In kannel send-sms URL need to add meta-data to send TLV parameter
meta-data will look like this
meta-data=?smpp?entityid=<Enrity ID>&templateid=<Template ID>

the valued should be URL encoded inside meta-data. So final meta-data will look like below

&meta-data=%3Fsmpp%3Fentityid%3D<Entity ID>%26templateid%<Template ID>

Note :- These all values and samples tested according to our environment . Please check the configuration according to your system and configuration.