BulkSMS API Document

API Version: 5.9
Document Version: 2.0
Last Updated Date: 31st July 2013

netCORE

Contents

Tables

netCORE

Chapter

1

Introduction

1.1 Purpose

This document provides a description of the **Bulk push API version 5.9** available for client use.

This document also describes the API for collecting delivery reports as well.

1.1 Scope

The scope of this document is limited to interaction

1.2 Audience

Enterprises who wish to integrate their existing software systems with the netCORE's BulkSMS API platform.

1.3 Document Organization

This section explains how information is distributed in this document. Table 1 presents an overview of the chapters in this document

Table 1-1: Chapter overview

#	Chapter	Contents
1	Introduction	Defines the purpose, scope, and intended audience of this document. Explains how information is organized in this document
2	API Overview	This chapter describes the details of the BulkSMS APIs.

1.4 Acronyms and Abbreviations

The acronyms and abbreviations expanded in Table 1-2 are fundamental to the information in this document.

Table 1-2: Acronyms and abbreviations used in this document

Acronym	Explanation
API	Application Programming Interface
CSV	Comma Separated Values
SMS	Short Messaging Service
HTTP	Hyper text transfer protocol
HTTPS	Hyper text Transfer protocol Secure
IP	Internet Protocol

1.5 Contact Details

For any queries related to this document please write to us at cs@netcore.co.in, or feel free to call us at +91-22-43664111.

Chapter

2

API Overview

This chapter contains an overview of netCORE's BulkSMS API.

2.1 Mobile number guidelines in API usage

For GSM & CDMA messaging, the Receiver Phone Number must start with country code - e.g. 91 in case of an Indian number.

No leading '0' or '+' are allowed—e.g., the number, 99860XXXXX, should be specified as 9199860XXXXX. This means the number should always be prefixed by 91 with no leading '+' or 0's.

For sending message internationally the mobile number should be prefixed with '00' the appropriate country code, followed by the mobile number. For all local Indian numbers, the mobile number must be 12-digit long. No special character like "-", "(",")" or anything similar is allowed in the phone number, e.g., 91-98123XXXXX is disallowed.

2.2 Choice between IP address & mobile number authentication

To be able to use our APIs the enterprise needs to choose between 2 access mechanisms -

1. White listing their public IP addresses –
   The series :- 10.0.0.0 through 10.255.255.255
   172.16.0.0 through 172.31.255.255 192.168.0.0 through
   192.168.255.255 are internal/private IP series and cannot be
   given for white listing.

2. Mobile number & password authentication. You need supply a Valid mobile number which will be registered for the API account. The account details with the password will be shared with you by CS team.

2.3 NDNC Settings for using API

In 2007, **Telecom Regulatory Authority of India** (**TRAI**) set up **National Do Not Call** (**NDNC,** also called as do not disturb, DND) **Registry** to curb unsolicited Commercial calls. i.e. any call/message, through telecom system promoting any commercial transaction for goods, investments or services, which a subscriber does not wish to receive.

It failed to help customers and in 2011, TRAI came up with new policy with name **National Customer Preference Register (NCPR)** and it became operational on 27th Sept, 2011

Mobile phone users who want to register their phone number for Do not call registry may **dial 1909** or **SMS to 1909** with keywords '**START 0**' for registration and '**STOP DND**' for de-registration. Requests will become effective in **7 days** from the date of request.

Partial Blocking: Apart from normal full blocking of telemarketing calls, subscriber now will also be able to opt for partially blocked mode, which will allow to receive promotions for specific categories like real estate, education, finance, entertainment, health and tourism etc. This seperate list will be now called as **National Customer Preference Register (NCPR)**. By default, all subscribers will be moved to fully blocked mode. For partial blocking, SMS needs to sent to 1909 will be 'START Option-Code' and 'STOP Option-Code' . Once changed, preference cannot be changed for next 3 months.

Option code for each category is provided below:

1. Banking/Insurance/Financial products/credit cards
2. Real Estate
3. Education
4. Health
5. Consumer goods and automobiles
6. Communication/Broadcasting/Entertainment/IT
7. Tourism and Leisure

So if you want to receive messages relating to only Banking products, then send SMS "START 1″ to 1909. Similarly, for receiving messages relating to Health and Education, send SMS "START 4,3″ to 1909.

How to complain: In case subscriber still receives call/messaes, he should complain to mobile service provider, mentioning the caller/sender telephone number, date/time of the call and details of message. All complaints must be actioned upon within 7 days.

If you receive any unsolicited commercial communications seven days after registration of your telephone number in the NCPR, you may register a complaint by:

1. Dialling the toll free number 1909; or
2. Sending an SMS to 1909

It is important to note that: The complaint has to be registered from the telephone number on which unsolicited commercial communication has been received. Your complaint must be made within three days of receipt of the unsolicited commercial communication.

2.4 SMS Header and Footer

Messages may need to be sent out with a fixed header and/ or footer in some cases. For instance, an Enterprise would want all messages to be signed-off with their name or all offers with Terms of Service/ Use. SMS Header and footer settings can be used for help in this regard.

If the header is set, all messages sent will be prefixed with the header content. Similarly, if the footer is set, all messages sent out via the API will end with the footer content.

For e.g : We have set the header of the message as _Dear Customer,‘ and the footer of the message as _Happy Shopping!‘. If the message text: _We have come up with a special offer for all our customer with a flat discount of 50% and more T&C apply. Visit your nearest store.‘ is specified in the API call, the message to the end recipient will be sent out as: _Dear Customer, We have come up with a special offer for all our customer with a flat discount of 50% and more T&C apply. Visit your nearest store. Happy Shopping!‘. The Header and Footer can be set from the BizBond dashboard from Keyword -> Message Customization.

2.5 API in Asynchronous mode

Messages can be sent out using Synchronous or Asynchronous mode. In Asynchronous mode, the API call will return immediately with the Request ID of the call – the complete input validation and message sending will be done asynchronously – this way the call made does not have to wait for the complete processing to be done. In the default mode / synchronous, the API will return a very detailed response XML; with the transaction ID, message ID for every message sent.

To use the API in the asynchronous mode, use the parameter: async =1.

Here‘s the sample XML returned in Asynchronous API mode:

```
<RESULT REQID ='15034672'>
</RESULT>
```

2.6 Expiry Time

Expiry time can be used to set a — validity period for the message. In case the message could not be sent out from the system within the time specified, for one of several reasons like – unexpected latency issues, heavy traffic, the message can be expired – if it is no longer relevant. This is extremely useful for time-critical messages. The expiry time parameter however does not include the time that it could actually take for a message to reach the user from the gateway/ operator/ network. The Expiry time needs to be specified in minutes.

Example: If the client wishes to send a time-critical message at 9:30 am, which needs to leave the system within 20 minutes, and reach users by 10:00 am; the expiry time could be set at 20 minutes – this way; in case the message gets delayed in the system, it will not be sent out at all.

2.7 Store and Forward

Using the Store and forward feature, enterprise clients can set a time when messages need to be buffered in netCORE system without being sent out as sms to end users; even if messages are being sent using the API during that interval. At the end of the "store" period, all the sms that were buffered in system from start of the "store" period till then will be sent out/ forwarded to the respective users.

Enterprises can specify a Store Start and End Time. Requests (API calls) for sending messages that come within the Store From and To period, will be sent out only after the Store End time.

For example:

Consider the case of an Enterprise whose back-end software sends messages continuously via the API irrespective of time of the day, but wishes to not disturb users with sms alerts at odd hours - Store and Forward feature is the ideal option here.

The Store Time can be set as 10pm - 8am.

All message requests via API that come in between 10 pm to 8 am will be buffered at system end. From 8 am, the next day the queued requests will be sent out to the users. Message requests that come in before 10pm or after 8 am will be sent to users immediately.

Store Time can also be set as 10am - 6 pm - to not send messages during working hours, for instance.

To enable this feature, please contact your Account Manager with the Store Start and End Time. This will be fixed per account and cannot be overridden via API. For any changes, you'll have to contact your Account Manager again.

2.8 Message Types

The most common message type that needs to be sent out by an Enterprise is a simple text message. The SMS API can be used to send WAP messages, flash message, Vcard message, as well as UDH. UDH and Binary messages can be sent using the Single Message API, Here‘s a quick reference guide on the advanced parameters that needs to be used in each of these cases.

Table 2-1: The different Message Types

Message Name	Message Type (mtype)	UDH (udh)	URL (url)	Text
Text (Default)	1	No	No	Yes
WAP	5	No	Yes	Yes (optional)
MULTILINGUAL	2	No	No	Yes
PICTURE	8	No	No	Yes (Hex Code)
VCARD	11	No	No	yes
Flash	12 (English); 15 (Multilingual)	No	No	Yes
UDH	13 or 14 (see UDH message type below)	Yes	No	Yes (optional)

With a brief example, outlining the sample steps to send UDH messages to a port number using Single Message API. In addition to the mandatory parameters as outlined in Section 2.9, specify the following:

Table 2-2: UDH Message Type – additional parameters

How to send UDH (port based) Message

Additional BulkApi URL parameter details for same

Parameter	Description
mtype	For English text **mtype**=13 and multilingual **mtype**=14
udh	User-defined data header. The data header is used for long messages as well as sending binary content. If you need to send message to specific port (for j2me application to receive), you may specify the information in udh parameters. When setting udh, mtype should be 13 or 14.
text	Text that needs to be sent on mobile handset. In case of binary content, see 'how to encode binary message'.

Step 1 : Prepare the UDH byte value with message size less than 140 byte, where application running on 5000 port.(hex value of 5000 = 1388)

06 05 04 **13** **88** 00 00

Step 2 : Make it 2-character percent-separated hex

%06%05%04%**13%88%**00%00

Step 3 : URL encode (UTF-8)

%2506%2505%2504%25**13**%25**88**%2500%2500

Step 4 : Use URL parameters &udh =%2506%2505%2504%25**13**%25**88**%2500%2500

2.3a UDH Octet parameter details

Octet Number	Value	Description
1	06	Length of the User Data Header
2	05	Information Element Identifier (IEI; application port addressing scheme,16-bit port address
3	04	Information Element Data Length (IEDL)
4-5	13 88	Information Element Data (octets 4 & 5 -> 13 88 - destination port)
6-7	00 00	Information Element Data (octets 6 & 7 -> 0000 - originator port)

2.3b Send Message to a port number with more than 140 bytes, need to add concatenated message header information

Octet Number	Value	Description
1	0B	Length of the User Data Header
2	05	Information Element Identifier (IEI; application port addressing scheme, 16-bit port address
3	04	Information Element Data Length (IEDL)
4-5	13 88	Information Element Data (octets 4 & 5 -> 13 88 - destination port)
6-7	00 00	Information Element Data (octets 6 & 7 -> 0000 - originator port)
8	00	Information Element Identifier (IEI; concatenated short message, 8-bit reference number)

9	03	Information Element Data Length (IEDL)
10	02	Information Element Data (concatenated short message reference number)
11	02	Information Element Data (total number of concatenated messages (0-255)
12	01	Information Element Data (sequence number of current short message)

How to encode Binary message

Step 1. 2-character percent-separated hex
Step 2. url-encode above
Step 3. &text=<output of step 2>

For more details about different octet in UDH header, Plz refer netcore's API document Table 2.2 & 2.3 section.

Note: To send normal text App running on destination port should be able to read ASCII characters. If app understands binary then message must be in binary encoded rather normal text.

2.9 API to send Single Message Using GET method

This API allows an enterprise to send a single message to one or more mobile numbers (Maximum of 50 Mobile numbers) at one time. This is achieved by issuing a HTTP GET request to our server listening at http://bulkpush.mytoday.com/BulkSms/SingleMsgApi with following GET parameters that must be URL encoded.

1. **feedid:** This is the unique identifier representing the channel owned by the enterprise/ the Account ID
2. **To:** Mobile number (12-digit) or comma separated mobile numbers
3. **Text**: Message to be sent to the users. The space within the text message will be replace with '+' sign.
4. **Username**: It is the 10-digit Mobile Number. This is optional parameter as explained in Section 2.2.
5. **Password**: It is the password will be received by SMS after getting registered on our platform. This is an optional parameter, mandatory only when the username is provided.

6. **Time:** Make sure that the time is specified in the correct format – yyyymmddhhmm. If time is not specified message will go immediately. If time is specified in wrong format it won't go out. If time parameter is empty or past time is mentioned then message will go immediately.

7. **senderid:** Using Sender-id is Optional. If there are single or multiple sender-ids then all the sender-ids have to be validated. For one message only one sender ID can be used. In case no Sender ID is mentioned in the API call then the default sender ID is attached to the message. Please speak with your account manager to set the default sender ID for your account – in case of single sender ID, it is automatically set as default.
8. **mtype:** Using mtype to indicate the type of message being sent is optional. The default setting is to send text message. Use mtype as outlined in Section 2.8 (1 – text; 5 – WAP; 8 – Picture message; 13 – UDH)
9. **url**: URL parameter needs to be specified for WAP Push type messages only. Send the WAP URL in this parameter
10. **udh**: The data header for UDH Type messages can be specified in this parameter, if message type UDH is chosen.
11. **jobname** : This is an optional parameter The name of the campaign can be provided in this parameter. It is used for future reference purpose

Following is a sample URL to be invoked for this feature.

```
http://bulkpush.mytoday.com/BulkSms/SingleMsgApi?feedid=1879&usernam
e
=9967025255&password=hello&To=919967025255&Text=Hello
check2350&time=200812110950&senderid=testSenderID
```

In the example above:

1. The Feed ID for this example is 13451 which must be replaced by the Feed ID/ Account ID for the channel owned by the client (provided by the Account Manager)
2. The 'To' parameter must contain either a single 12-digit mobile number or multiple mobile numbers which are comma separated.
3. The parameter senderid is optional – and it should only contain one of the white listed sender IDs only. If otherwise then there will be an error.
4. The data in the 'Text' parameter has to be URL encoded.

5. username is the mobile number which is registered on Netcore Platform.

6. password will be received a message after getting registered on our platform.
7. time parameter has the value 11th Dec, 0950 hrs

8. senderid parameter has value testSenderID

The success or failure details are returned as an XML.

If the message format is correct and our servers have accepted the API call, then a message of the following format is sent back:

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID ='6'>
    <MID SUBMITDATE='2008-06-04 19:10:41' ID='1' TAG = 'null' TID =
'9'>
    </MID>
</RESULT>
```

REQID is the Request ID, TID is the Transaction ID – here the request has a single Transaction ID. Transaction ID is assigned per text message. In the example above – single text message was sent and hence single transaction ID. You can query for Delivery Reports using Transaction ID (TID) & Request ID (REQID) also. The use of ID & TAG is present for XML request-response scenarios just to adhere to the output DTD.

Eg of flash message:

http://bulkpush.mytoday.com/BulkSms/SingleMsgApi?feedid=123456&username=9320925174&password=a sdad&To=9920694669&Text=testing%20flash&mtype=12

Result O/P

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID ='1919519053'>
   <MID SUBMITDATE='2013-07-03 10:33:20' ID='1' TAG = 'null' TID =
'3582537506'>

   </MID>
</RESULT>
```

E.g. of vcard message:

http://bulkpush.mytoday.com/BulkSms/SingleMsgApi?feedid=123456&To=9920694668&mtype=11&Text=name=TEST,mobile=9849895645&username=9952166441&password=sdfsdf

2.9.1 Using POST method

This API usage is different than the previous one as it allows you to upload up-to 10,000 messages.

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>Single Message API Page</title>
    < /hea
    d>  <
    bod y>
        <h1><center>Single Message API</center></h1>
        <form method="post"
action="http://bulkpush.mytoday.com/BulkSms/SingleMsgApi">  <table>
                <tr><td>
                Feed id </td><td><input type="text" name="feedid"
value="" size="20"></td></tr>
                <tr><td>
                      Sender id (Optional) </td><td><input type="text"
name="senderid" value="" size="20"></td>
                </tr>
                <tr><td>
                  Username (optional)</td><td><input
type="text" name="username" value="" size="20"></td></tr>
                <tr><td>
                Password (optional)</td><td><input type="password"
name="password" value="" size="20"></td></tr>
                <tr><td>
                            Future Publish
                                Time  (
Optional)<br>(yyyymmddhhmm)<br>Ex:200908271351</td><td><inp ut
type="text" name="time" value="" size="20"></td>
                </tr>  <tr>
                       <td> Mobiles Number(Comma
separated)</td><td><input type="text" name="To" value=""
size="100"></td>
                </tr>
                <tr><td>
                       Message</td><td><textarea name="Text" value=""
```

```
          rows="4" cols="100" ></textarea></td>
                          </tr>
                          <tr><td></td><td><input type="submit" name="Ok"
          value="Go"> <input type="reset" name="reset" value="Reset">
                          </td></tr>
                       </table>
                    </form>
                </body>
          </html>
```

```
Example: Mobile no: 919900840066,919900840067

Message : hello welcome to bulk push ......
```

In the above post method

1. 'To' parameter must contain either a single 12-digit mobile number or multiple mobile numbers which are comma separated.
2. Feed ID/ Account ID for the channel owned by the client (provided by the Account Manager)
3. senderid (Optional) can be given within double quotes.
4. time (Optional) correct format – yyyymmddhhmm. If time is not specified message will go immediately. If time is specified in wrong format it won't go out.

Sample Error messages:-

In case the Enterprise Account/ Channel has not been registered with us:

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT>
<REQUEST-ERROR>
<ERROR>
<CODE>102</CODE>
<DESC>Client 18244 is not registered</DESC>
</ERROR>
</REQUEST-ERROR>
</RESULT>
```

In case the credits have expired/ are insufficient in the Enterprise Account:

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT>
<REQUEST-ERROR>
<ERROR>
<CODE>104<CODE>
<DESC> Credit not sufficient </SC>
</ERROR>
</REQUEST-ERROR>
</RESULT>
```

Table 2-4: The different error codes in the system

Error Code	DESC-String	Scope	Example	Description
101	XML MALFORMED	REQUEST	XML MALFORMED. The end-tag for element type "TEXT" must end with a '>' delimiter.	XML does not match with required DTD
102	Client < Account ID> is not registered	REQUEST	Client 18888 is not registered	If the Account id for the Client is not registered with MyToday
103	Login failed	REQUEST		Invalid login information entered
104	Credit is insufficient	REQUEST	Credit is insufficient	If the credit for Account expired
105	<IP address> is not white listed	REQUEST	127.40.0.12 IP is not white-listed	If the IP of calling client is not registered for Account
106	Message Empty	MESSAGE		When the Message text is blank
107	Message Id not unique	REQUEST		Mid is not Unique in submitted XML
108	Mobile number <Mobile Number > invalid	SMS	Mobile number 91-9322464675 invalid	If invalid mobile number specified
109	Sender id <sender id> not registered for account <Account id>	SMS		Sender ID needs to be registered
110	Indexes not unique	MESSAGE		If a message having two recipient having same index number

111	<Mobile> User not subscribed to this account	SMS	99900840096 User not subscribed to this account	
112	Message id <id Number > is not an Integer	MESSAGE	Message id Hello is not an Integer	
113	Index <index Number> is not an integer	MESSAGE	Index Two is not an integer	Index number of recipient is non numeric
114	<Sender Id> sender id is invalid	MESSAGE	12@3 sender id is invalid	If sender id Contains special character except Dot(.) and Alpha numeric greater than 11 char and Numeric > 16 digit
115	Please check that the tag is not greater than 25 characters or contains special characters	MESSAGE		
116	Incorrect Version number	MESSAGE	Incorrect Version number	Other than 1.0
117	No registered sender id	MESSAGE		
118	Long message feature not enabled	MESSAGE		If message text is greater than 160 characters and for multi lingual more than 70
119	Multi lingual feature not enabled	MESSAGE		If message text contains other than English and client has been not enabled multi lingual feature
120	International SMS feature not enabled	MESSAGE	09900000096	If mobile is greater or lesser than 12 digit number and International feature has been not enabled
121	Account id <Account Id>invalid	MESSAGE	Account id India invalid	If Account id is non numeric
122	User name is empty.	REQUEST		If user does not supply in case user name required
123	Password is empty.	REQUEST		If user does not supply in case password required
124	User name <User	REQUEST	User name 9900840000 has not been configured. Please check with Account	

	name>not configured, Please check with Account Manager.		Manager.	
125	Invalid User Name and password	REQUEST	user name: {9886307244} and password do not match	If specified user name and password mismatch
126	Access denied User <User name> does not have rights to use the API. Please check with Account Manager.	REQUEST	Access denied User 9900840000 does not have rights to use the API. Please check with Account Manager.	
127	Conflict, more than one BULKPUSHAPI role exists for user <User name>	REQUEST	Conflict, more than one BULKPUSHAPI role exists for user 9900840000.	Role configuration incorrect. If you get this Error - Contact your Account Manager.
128	Access denied to URL <URL> for user name <User name>. Please check with Account Manager.	REQUEST	Access denied to URL http://XXX/BulkSms/UploadCsvFile for user name 9900840000. Please check with Account Manager.	
129	Invalid Time	REQUEST	Correct Time is not been specified 20080930130	
130	Invalid Expiry Time	REQUEST	If expiry time is not numeric in database Example: a124	Expiry time configuration incorrect. . If you get this Error - Contact your Account Manager.
131*	NDNC Status is currently unavailable	REQUEST	Message to user 91990084xxxx rejected, since the NDNC Status is currently unavailable. This means that MyToday has not yet checked with the NDNC registry about the status.	This happens when the system is unable to get the NDNC status for a particular mobile number.
132*	The user is registered for NDNC	REQUEST	Message to user 91990084xxxx rejected, since the user is registered for NDNC and it means that user does not want to receive any unsolicited messages	This happens when the user is registered with NDNC & doesn't wish to receive sms.
136*	The user NDNC status unknown	REQUEST	Message to user 91990084xxxx rejected, since the user NDNC status unknown. This is the case when NDNC registry does not know the status of the mobile number. This happens when a number is in the market while NDNC is not available of its existence.	This happens when the NDNC status is not available for that particular user
-1	Invalid Record Format	REQUEST	Feed id is empty , Mobile number is empty	If record does not full fill required standard

*The error codes 131, 132, 136 occur only if the channel is set-up for NDNC validation.

Please find below few more error codes that can occur

137	There is an error in the account setting. Please contact your Account Manager
138	Error occurred while contacting payment portal ,Please try after some time or write us to mbiz@mytoday.com
139	Sorry Account not exist ,Please contact account manager or write us to mbiz@mytoday.com
140	Invalid message transaction id,Please give greater than %s -- Deprecated
141	Message type [%s] invalid or Not supported
142	URL parameter [%s] is empty for message type [%s]
143	Ambiguity,For Message type [%s] Can not have [%s] data
144	Message type [%s] Not supported to DIY user [%s]
145	Found duplicate Request with msgid %s,Request dropped
146	Message to user %s rejected,Since message can not be schedule after 30 days
147	User %s scheduled for NDNC hibernate has been expired -- Deprecated
148	%s is blacklisted senderid
149	%s is not whitelist virtual senderid
150	Promotional Message to user %s rejected ,Since publish or schedule hour %s falls between TRAI Do not call
151	Request_id- %s,Txid- %s, Senderid- %s, Message to user- %s dropped due to insufficient balance
152	Duplicate mobile number [%s] for a message

2.10 API to send custom messages to multiple numbers using CSV

The API allows an enterprise to upload multiple messages at a time by creating a file containing mobile numbers and the corresponding messages (either same message to everyone or different messages).

2.10.1 CSV uploads using '*filestring*' parameter

2.10.1.1 Using the GET HTTP call

A CSV with mobile numbers in column 1 and the corresponding messages in column 2 needs to be created. Please note that even when the same message is being sent to all numbers, the messages must be copied and paste against every number.

The CSV contents can be directly specified in the parameter

http://bulkpush.mytoday.com/BulkSms/UploadCsvFile

The parameters to this API are:

1. **feedid:** This is the unique identifier representing the channel owned by the enterprise/ the Account ID
2. **filestring:** Mobile Number followed by the message to be sent, separated by a comma. Messages to different mobile numbers will be in new-lines. The content should be URL-encoded. Each number-message combination comes in a newline and each newline needs to be URL encoded. Each text message begins & ends in double quotes – which in turn is URL encoded as %22. In case a double quote has to be included in the text, then this needs to be escaped using a double quote itself. Hence a message Hello||World will be written as Hello||‘World.
3. **username**: It is the 10-digit Mobile Number. This is optional parameter.
4. **password**: It is the password will be received by SMS after getting registered on our platform. This is an optional parameter, mandatory only when the username is provided.
5. **Time:** Make sure that the date is specified in the correct format – yyyymmddhhmm. If time is not specified message will go immediately. If time is specified in wrong format it won‘t go out.

FeedID must be set to the Channel Identifier allocated to the enterprise and filestring is the actual content - containing the numbers and messages to be sent. It also has the optional Sender ID in double quotes comma separated after the message ends. There can be more than one sender IDs available for one channel – but these need to be approved on the network. Please contact your account manager to discuss more on this.

If no sender ID is specified in the API then the default sender ID for this channel will be picked. Hence using sender ID in the API is optional. Please note, the keyword/ title of the Channel will not appear anywhere in the text of the messages. The URL has a maximum limit of 2048 character.

A sample API call in HTTP GET format -

```
http://bulkpush.mytoday.com/BulkSms/UploadCsvFile?feedid=38244&username=&password=&time=200910201713&filestring=919900840069+%2C%22Test++message+1%22%2C%22Super%22%0D%0A919900840070+%2C%22Test++message+2%22%2C%22Super%22%2C%222009101519 25%22
```

If you notice, the string is URL encoded. The message itself needs to be specified in double quotes. The sender ID also needs to be specified in double quotes. Between 2 successive messages, there should be a new- line.

The success/failure details are returned as an XML.

If the message send is successful, you‘ll get a message of the format:

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
```

```
        <RESULT REQID ='7'>
                <MID SUBMITDATE='2008-06-05 09:16:46' ID='1' TAG = '' TID =
        '11'>
</MID>
 <MID SUBMITDATE='2008-06-05 09:16:46' ID='2' TAG = '' TID =
        '12'>
                </MID>
        </RESULT>
```

REQID is the Request ID, TID is the Transaction ID – here the request has a 2 Transaction IDs. As mentioned in Section 2.2 Transaction ID is given per text message. In the example above – there are 2 text messages and hence 2 transaction IDs in this response XML. You can query for Delivery Reports using Transaction ID or Request ID also.

ID & TAG are present for use in XML request-response. These cannot be used for querying for Delivery Status reports right now.

NOTE: You can also specify a Future time for the entire file upload – the time can be specified in the —time‖ parameter – this will apply to all the messages in the file, unless there's a specific time associated with a message in a file.

2.10.1.2 Using the POST HTTP call

In this format the user has to provide with their feed-id and then mobile numbers and text message separated by a comma and text of the message within the double quotes. The optional sender ID is again comma separated after the message and is also within double quotes.

This API usage is different than the previous one as it allows you to upload up-to 10,000 messages.

Sample piece of code that shows how to use this API in POST HTTP call-

```
<html>
   <head>
         <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
       <title>CSV String Page</title>
   </head>  <body>
            <h1><center>CSV String API</center></h1>
        <form method="post"
action="http://bulkpush.mytoday.com/BulkSms/UploadCsvFile">


           <table>
               <tr><td>  Feed id
</td><td><input type="text" name="feedid" value=""
size="20"></td></tr>
        <tr><td>
        Username (optional)</td><td><input type="text"
name="username" value="" size="20"></td></tr>
            <tr><td>
          Password (optional)</td><td><input type="password"
name="password" value="" size="20"></td></tr>
        <tr><td>
           Future global time(optional)</td><td><input type="text"
name="time" value="" size="20"></td></tr>
                            <tr><td>
               CSV String</td><td><textarea name="filestring"
rows="10" cols="100"></textarea></td></tr>
               <tr><td></td><td><input type="submit" name="Ok"
value="Go"> <input type="reset" name="reset" value="Reset"> </td></tr>
           </table>
</form></br> </body>
</html>


Examples: 919900840000 ,"Test message 1" ,"Optional sender id
","Optional Future Time"
```

this method the enterprise can use different sender-id for different mobile number

In

Enterprises can specify a different sender ID per message sent using the Optional sender ID parameter

On message specification: if you want to send double quotes within a message then specify the double quotes as two consecutive double quotes For example, if you want to send message He said "wow!" to harry

the message should be specified as:
He said ""wow!"" to harry

NOTE: All the records in the file don't have to have the same format. Example, in one record future time could be specified, in another it can be missed (then message will be sent immediately or at the global future time specified in the main API call), Sender ID is also optional in the same way.

2.10.2 CSV uploads in the file format

This API can only be called using HTTP POST – as the file format CSV is not understood by GET mechanism.

The API that lets you upload a CSV file through a POST HTTP call is the following –

http://bulkpush.mytoday.com/BulkSms/UploadFormFile

A sample HTML form code to show you how to invoke this API is as following. Post HTTP can be used to send up-to 10,000 Numbers –

```
<html>
<head>
     <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
    <title>CSV File Upload Page</title>
</head>  <body>
          <h1><center>Upload CSV File</center></h1> <form
method="post"  enctype="multipart/form-data"
action="http://bulkpush.mytoday.com/BulkSms/UploadFormFile">
        <table>
            <tr><td>  Feed id
</td><td><input type="text" name="feedid" value=""
size="20"></td></tr>      <tr><td>
            Username (optional)</td><td><input type="text"
```

```
name="username" value="" size="20"></td></tr>
                <tr><td>
                Password (optional)</td><td><input type="password"
name="password" value="" size="20"></td></tr>
        <tr><td>
                Future publish Time (optional)</td><td><input
type="text" name="time" value="" size="20"></td></tr>
                <tr><td>
                Select CSV File </td><td> <input type="file"
name="upload" value="" width="30"/></td></tr>
<tr><td></td><td><input type="submit" name="Ok" value="Go"> <input
type="reset" name="reset" value="Reset">
                </td></tr>  </table>
       </form>
       <br>  </body>
</html>
```

The file content will be of the form:

—MobileNo||,"Message Text","Optional sender id","Optional Future Time"

Example:919900840001,"Test Message","SenderID1",”200901010900”

2.10.3 Sample Error messages

In case the Enterprise Account/ Channel has not been registered with us:

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT>
<REQUEST-ERROR>
     <ERROR>
          <CODE>102</CODE>
          <DESC>Client 18244 is not registered</DESC>
     </ERROR>
 </REQUEST-ERROR>
 </RESULT>
```

In case the credits have expired/ are insufficient in the Enterprise Account:

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT>
<REQUEST-ERROR>
     <ERROR>
          <CODE>104<CODE>
            <DESC> Credit not sufficient </DESC>  </ERROR>
```

```
</REQUEST-ERROR>
</RESULT>
```

2.11 API to send custom (different) messages to multiple numbers via XML

This API allows an enterprise to send multiple messages to users via XML. Same message can be specified for all users, or different messages could be specified to different users.

Different Sender ID can be specified for different messages – the Sender IDs need to be white-listed with Netcore prior to use – else the default Sender ID set for the enterprise account will be used to send messages to users.

The base API is:

http://bulkpush.mytoday.com/BulkSms/SendSms

The parameter to this API that should be submitted as a POST request is:

UserRequest: The XML file containing the account and message details. The XML File must be submitted as The XML file must be URL- encoded and XML-encoded.

2.11.1 Using GET method

We show here a sample piece of code where this API is being used through the GET method of HTTP call. Note that the parameter UserRequest takes the XML input which

```
http://bulkpush.mytoday.com/BulkSms/SendSms?UserRequest=%3C%21DOCTYPE
+REQ+SYSTEM+%27http%3A%2Fbulkpush.mytoday.com%2F%2FBulkSms%2FBulkSmsV
1.00.dtd%27%3E%0D%0A%3CREQ%3E%0D%0A%3CVER%3E1.0%3C%2FVER%3E%0D%0A%3CU
SER%3E%0D%0A++++++++%3CUSERNAME%3E%3C%2FUSERNAME%3E%0D%0A++++++++%3CP
ASSWORD%3E%3C%2FPASSWORD%3E%0D%0A%3C%2FUSER%3E%0D%0A%3CACCOUNT%3E%0D%
0A++++++++%3CID%3E38272%3C%2FID%3E%0D%0A%3C%2FACCOUNT%3E%0D%0A%3CMESS
AGE%3E%0D%0A++++++++++++++++%3CTEXT%3Ehello+hi%3C%2FTEXT%3E%0D%0A++++
++++++++++++++++++%3CTYPE%3E0%3C%2FTYPE%3E%0D%0A++++++++%3CMID%3E1%
3C%2FMID%3E%0D%0A++++++++++++++++%3CSMS+FROM%3D%27%27+TO%3D%279199008
40069%27+INDEX+%3D%271%27%3E%3C%2FSMS%3E%0D%0A%3C%2FMESSAGE%3E%0D%0A%
3CMESSAGE%3E%0D%0A++++++++++++++++%3CTEXT%3Ehello%3C%2FTEXT%3E%0D%0A+
++++++++++++++++++++%3CTYPE%3E0%3C%2FTYPE%3E%0D%0A++++++++%3CMID%3
E2%3C%2FMID%3E%0D%0A++++++++++++++++%3CSMS+FROM%3D%27%27+TO%3D%279199
00840069%27+INDEX+%3
D%271%27%3E%3C%2FSMS%3E%0D%0A%3C%2FMESSAGE%3E%0D%
0A%3C%2FREQ%3E%0D%0A
```

XML file that was URL encoded above for GET call of the API

```
<!DOCTYPE REQ SYSTEM
'http://bulkpush.mytoday.com//BulkSms/BulkSmsV1.00.dtd'>
<REQ>
<VER>1.0</VER>
<USER>
        <USERNAME>99670xxxxx</USERNAME>
        <PASSWORD>abcde</PASSWORD>
         </USER>
         <ACCOUNT>
        <ID>38272</ID>
        </ACCOUNT>
        <MESSAGE>
```

```
    <TEXT>hello hi</TEXT>
    <TYPE>0</TYPE>
    <MID>1</MID>
    <SMS FROM='' TO='919900840069' INDEX ='1'></SMS>
</MESSAGE>
<MESSAGE>
     <TEXT>hello</TEXT>
     <TYPE>0</TYPE>
     <MID>2</MID>
     <SMS FROM='' TO='919900840069' INDEX ='1'></SMS>
</MESSAGE>
</REQ>
```

Output for the given above given sample of API call using GET method -

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID ='3738'>
       <MID SUBMITDATE='2008-09-26 16:47:38' ID='1' TAG = 'null' TID =
'574795'>
       </MID>
       <MID SUBMITDATE='2008-09-26 16:47:38' ID='2' TAG = 'null' TID =
'574796'>


       </MID>
</RESULT>
```

2.11.2 Guidelines to create messages

While creating the messages – all special characters need to be escaped. Do find the table below that describes the conversion of these special characters.

Table 2-5: Special characters conversion

Code Character	**Replace with**
Tab(\t)	

Single Quote(')	'
Double Quote (—)	"
New Line(\n)
Ampersand(&)	&
Percentage (%)	%
Greater than(>) Lesser than(<) \f	> <

You may use the same conversion if you choose to insert special characters in the tag parameter – but we recommend that this should not be done.

2.11.3 DTD for input XML

The DTD of the XML with mobile-message-sender ID input is as follows:

```
<!ELEMENT REQ (VER,USER?,ACCOUNT,MODE?,NET-MODE?,MESSAGE+)>
<!ELEMENT MODE (#PCDATA)>
<!ELEMENT NET-MODE (#PCDATA)>
<!ELEMENT VER (#PCDATA)>
<!ELEMENT USER (USERNAME?,PASSWORD?)>
```

```
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT PASSWORD (#PCDATA)>
<!ELEMENT ACCOUNT (ID)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT MESSAGE
(TAG?,TEXT?,DLR?,TYPE?,MID,VALIDITY?,HIBERNATE?,URL?,UDH?,SMS+)>
<!ELEMENT TEXT ANY>
<!ELEMENT TAG ANY>
<!ELEMENT MID (#PCDATA)>
<!ELEMENT DLR (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT VALIDITY (#PCDATA)>
<!ELEMENT HIBERNATE (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT UDH (#PCDATA)>
<!ELEMENT SMS EMPTY> <!ATTLIST
SMS FROM CDATA #IMPLIED
    TIME CDATA #IMPLIED TO
    CDATA #REQUIRED INDEX
    CDATA #REQUIRED TAG
    CDATA #IMPLIED
>
```

2.11.4 Sample Input

Here‘s an example to understand the input format better

```
<!DOCTYPE REQ SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsV1.00.dtd'>
<REQ>
   <VER>1.0</VER>
   <USER>
        <USERNAME>99670xxxxx</USERNAME>
        <PASSWORD>abcde</PASSWORD>
   </USER>
   <ACCOUNT>
       <ID>3146</ID>
   </ACCOUNT>
   <MESSAGE>
       <TAG>Morning Greet</TAG>
    <TEXT>Have a Great Day, Everyone!</TEXT>
       <DLR>1</DLR>
       <TYPE>0</TYPE>
       <MID>1</MID>
     <VALIDITY>0</VALIDITY>
       <SMS FROM = 'MyToday' TO = '919967025255'
```

```
TIME='200806010900' INDEX = '1' TAG = ''></SMS>
<SMS FROM = 'BizBond' TO = '919956789541' TIME='200806010900' INDEX =
'2' TAG = ''></SMS> <SMS FROM = 'Netcore' TO = '919957789541'
TIME='200806010900' INDEX =
'3' TAG = ''></SMS>
   </MESSAGE>
   <MESSAGE>
       <TAG>Morning Reminder</TAG>
       <TEXT>Company Meeting scheduled for 11:30 am in the Conference
Room. Please attend</TEXT>
       <DLR>1</DLR>
       <TYPE>0</TYPE>
       <MID>2</MID> <VALIDITY>0</VALIDITY> <SMS
       FROM = 'BIZ A' TO = '919967025255'
TIME='200806010900' INDEX = '1' TAG = ''></SMS>
<SMS FROM = ' BIZ A ' TO = '919956789541' TIME='200806010900' INDEX
= '2' TAG = ''></SMS>
<SMS FROM = ' BIZ A' TO = '919956889541' TIME='200806010900' INDEX =
'3' TAG = ''></SMS>
<SMS FROM = ' BIZ A' TO = '919956889595' TIME='200806010900' INDEX =
'4' TAG = ''></SMS> </MESSAGE>
</REQ>
```

Table 2-6: Explanation of the different XML Tags

Element	Description	Mandatory	Default value	Examples
VER	Version	Yes	None, should be 1.0 for this initial release	1.0
USERNAME	User name of client-ten digit mobile number	No	None	996766666
PASSWORD	Password corresponding to the above user name	No	None	ourtomorrow

TYPE	Message Type 0 – Plain Text SMS 1 – Flash SMS And so on. Currently only plain text sms is supported	No	0	0
ACCOUNT	The account details of the user	Yes	None	
MODE	This specifies the level at which we perform subscriber checking 0. Default no checks performed for the receiver of the sms . 1. is the receiver subscribed to the account id or not.	No	0	0
ACCID	The account id. It is against this id that we validate the Sender ids, credits and other attributes. Ids should be valid integers	Yes	None	1410
MESSAGE	Encapsulates a content block. Multiple MESASGE blocks are allowed.	Yes	None	
TAG	A text that identifies the message. This text is supplied by the client. The response will contain this tag along with the message id. Minimum 7 chars and max 25 chars if present. Allowed chars [A-Z a,z 0-9, _, <space>]	No	None	Winner message
ID	Unique ID of message supplied by the client. The client sends this value. This has to be unique per MID in a request. The server sends this value back to the client in response to the client request. Each MESSAGE tag must contain an unique value for ID within the request	Yes	None	1234
TEXT	The message we have to send to the mobile number(s). Please see the Section 2.3 for the list of characters to be escaped	Yes	None	Congratulations on winning the AUDI TT.
DLR	This flag indicates whether the service shall ask for delivery reports or no. 0 is NO DLR required 1 is Yes DLR is required.	No	0	1
TYPE	This flag is used to indicate the Message type that needs to be sent: (1 – text; 5 – WAP; 7 – Mono ringtone; 10 – Non-Nokia mono ringtone; 8 – Picture message; 13 – UDH). For details, refer to Section 2.8	No	1	5 (to send WAP messages)
VALIDITY	Set this validity of a message to current SMSC time plus minutes specified in validity field. SMSC will not try to send the message after the validity have expired. 0 is infinite validity	No	0	10

HIBERNATE	If your Account is NDNC-Check enabled, messages will be sent out only to users who are not registered with NDNC. The platform will also reject numbers for whom the status is unavailable at the point of message send – to retry these messages when mobile NDNC Status is available, set this attribute to 1	No	0	1 to retry sending messages that were rejected due to NDNC status unavailability or expiry at the point of sms send
URL	Specify the URL that needs to be sent out in the WAP Push message using this	No	-	http://mytoday.com
UDH	Use this parameter to specify the user-defined header while sending message to an application port number	No	-	%2506%2505%2504%2515%2582%2500%2500
SMS	This tag contains the details of the person receiving the sms the attributes are explained below	Yes	None	No values. All values are supplied in the attributes
FROM	The sender of the message	No	Default sender id registered with us	MyToday
TO	Person receiving the SMS. See section 2.2 to know more about Mobile number guidelines	Yes	None	919967025255
INDEX	Unique Sequence ID. Must be an integer and must be unique to each recipient. While checking message status you must send this value.	Yes	None	Any integer like 1, 154, 3004
TAG	A text that identifies the recepient. This text is supplied by the client. Minimum 7 chars and max 25 chars if present. Allowed chars [A-Z a,z 0-9, _, <space>]	No	None	Winner message
TIME	Time at which the message needs to be sent. Messages can't be scheduled for future yet. If time specified is less than current time, current time is assumed. Time Format should be yyyymmddhhmm	No	Current Time	200802011700

2.11.5 DTD for output XML

The XML API call returns an XML Response which acknowledges the receipt of the request.

```
<!ELEMENT CODE ( #PCDATA ) >
<!ELEMENT DESC ( #PCDATA ) >
<!ELEMENT ERROR ( CODE?, DESC?, ERROR* ) >
<!ELEMENT REQUEST-ERROR ( ERROR ) >
<!ATTLIST ERROR INDEX NMTOKEN #IMPLIED >
```

```
<!ELEMENT MID ( ERROR* ) >
<!ATTLIST MID ID NMTOKEN #REQUIRED >
<!ATTLIST MID SUBMITDATE CDATA #REQUIRED >
<!ATTLIST MID TAG NMTOKEN #IMPLIED >
<!ATTLIST MID TID NMTOKEN #IMPLIED >
<!ELEMENT RESULT ( REQUEST-ERROR|MID+ ) >
<!ATTLIST RESULT REQID NMTOKEN #IMPLIED >
```

The format of the response is as follows - RESULT tag will always be present. The tag REQUEST-ERROR will follow if it's a global error like invalid request. **Table 2-7: Explanation of the different output XML Tags**

Element	Description
CODE	Numeric value – when an error occurs – this contains the numeric code for the error
DESC	This is the text description of the error
MID	Message ID as mentioned in the input XML,
ID	ID for the record as mentioned in the input XML
TID	In case of successful handling this is the transaction ID as returned by our platform. The same can be used to later fetch the delivery reports for the records processed.

In the following example, the request had multiple errors.

```
<!DOCTYPE REQ SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsV1.00.dtd'>
<REQ>
   <VER>1.0</VER>
   <USER>
        <USERNAME>99670xxxxx</USERNAME>
        <PASSWORD>abcde</PASSWORD>
   </USER>
   <ACCOUNT>
       <ID>31456</ID>


  </ACCOUNT>
  <MESSAGE>
      <TAG>Morning Greet</TAG>
   <TEXT>Have a Great Day,π Everyone!</TEXT>
      <DLR>1</DLR>
      <TYPE>0</TYPE>
      <MID>1</MID>
    <VALIDITY>0</VALIDITY> <SMS FROM = 'MyToday' TO =
'996025255' TIME='200806010900' INDEX = '1' TAG = ''></SMS>
```

```
<SMS FROM = 'BizBond' TO = '919956789541' TIME='200806010900' INDEX =
'2' TAG = ''></SMS> <SMS FROM = 'Netcore' TO = '919957789541'
TIME='200806010900' INDEX =
'3' TAG = ''></SMS>
   </MESSAGE>
   <MESSAGE>
       <TAG>Morning Reminder</TAG>
       <TEXT>Company Meeting scheduled for 11:30 am in the Conference
Room. Please attend</TEXT>
       <DLR>1</DLR>
       <TYPE>0</TYPE>
       <MID>2</MID> <VALIDITY>0</VALIDITY>
       <SMS FROM = 'BIZ A' TO = '919967025255' TIME='200806010900'
INDEX = '1' TAG = ''></SMS>
<SMS FROM = ' BIZ A ' TO = '919956789541' TIME='200806010900' INDEX
= '2' TAG = ''></SMS>
<SMS FROM = ' BIZ A' TO = '919956889541' TIME='200806010900' INDEX =
'3' TAG = ''></SMS>
<SMS FROM = ' BIZ A' TO = '919956889595' TIME='200806010900' INDEX =
'4' TAG = ''></SMS> </MESSAGE>
</REQ>
```

Message ID - MID 1 had invalid characters. Also the mobile number specified in index 1 was not valid. Because this MID did not have valid text, which is an attribute of the MID it did not generate a unique Transaction ID - TID from the server.

MID 2 in the same request was valid and hence was successfully submitted for delivery. The Transaction ID returned - TID 175432414 can be used to query for Delivery status.

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID=876565>
<MID SUBMITDATE= "200803121030" ID="1" TAG="Morning Greet" >
    <ERROR >
    <CODE>119<CODE>
    <DESC> Multi lingual feature not enabled</DESC>
</ERROR>
    <ERROR INDEX="1">
          <ERROR>
                <CODE>120<CODE>
                <DESC> International feature not enabled</DESC>
          </ERROR>
    </ERROR>
</MID>
```

```
        <MID SUBMITDATE= "200803121030" ID="2" TAG="Morning Reminder" TID=
"175432414" >
        </MID>
        </RESULT>
```

For each request submitted, the API will return a Request ID - REQID as an attribute of the result element. In the above example REQID 876565 is a unique integer that is returned from the API in response to the send message API call.

One unique Message ID - MID for each <MESSAGE> </MESSAGE> tag in the request is also returned. Each unique MID will have 1 unique system generated Transaction ID - TID that will be returned from the Bulk Push API. Transaction ID is generated only when there are no errors detected by the Message Send API. If a Transaction ID is returned, the message can be considered as ―successfully‖ submitted for Delivery. This Transaction ID can be used by the client for requesting delivery status.

2.11.6 Sample Output

Case 1

This Response XML means that all the messages have been handled successfully as there is transaction ID attributes within the MID. Receiving TID means the messages have been accepted by our platform successfully for further delivery.

```
        <!DOCTYPE RESULT SYSTEM
        'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
            <RESULT REQID ='3004'>
               <MID SUBMITDATE='2008-08-06 18:58:15' ID='1' TAG = 'null'
TID='553307'>
               </MID>
               <MID SUBMITDATE='2008-08-06 18:58:15' ID='2' TAG = 'null' TID =
        '553308'>

                </MID>
                <MID SUBMITDATE='2008-08-06 18:58:15' ID='3' TAG = 'null' TID
        = '553309'>

                </MID>
                </RESULT>
```

Case 2

This Response XML means that there was error in the formation of the complete input XML.

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>

<RESULT>
        <REQUEST-ERROR>
                <ERROR>
  <CODE>-1</CODE>
<DESC>Invalid record format</DESC>
                </ERROR>
        </REQUEST-ERROR>  </RESULT>
```

Case 3

This response XML means that some Messages were correctly formatted while some had error in the input XML.

ID-1:This had the potential error that your feed was not set to send international sms while the input XML was trying to send a sms to an international number- hence the error code 120 – international sms not supported

ID-2: The MID gets an TID(Transaction ID) attribute means the messages where accepted by our platform successfully for further delivery.

ID-3: This MID record in the input XML had the message node as empty – which is not allowed. As no empty SMS needs to go through the channel – hence the error code 106.

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID ='3020'>
        <MID SUBMITDATE='2008-08-07 11:57:46' ID='1' TAG = 'null'>
                <ERROR>
  <CODE>120</CODE>
<DESC>International sms feature not  enabled</DESC>
  </ERROR>
        </MID>
        <MID SUBMITDATE='2008-08-07 11:57:46' ID='2' TAG = 'null' TID =
'553336'>

        </MID>
        <MID SUBMITDATE='2008-08-07 11:57:46' ID='3' TAG = 'null'>
                <ERROR>
```

```
    <CODE>106</CODE>
    <DESC>Message Empty</DESC>
</ERROR>
                </MID>
                </RESULT>
```

2.11.7 Sample scenarios with error responses

2.11.7.1 Request level error

Here are few examples of a Request that failed at the Request level itself. Note that in this example the error failed at the request level.

```
<!DOCTYPE RESULT SYSTEM
   'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
   <RESULT>
   <REQUEST-ERROR>
          <ERROR>
               <CODE>104<CODE>
                <DESC> Credit not sufficient </DESC>
         </ERROR>
   </REQUEST-ERROR>
   </RESULT>
```

The error that is returned, if the XML format/ input format is incorrect

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT>
<REQUEST-ERROR>
<ERROR>
        <CODE>101</CODE>
        <DESC>XML MALFORMEDnull</DESC>
</ERROR>
</REQUEST-ERROR>
</RESULT>
```

2.11.7.2 Client not registered

INPUT

```
<!DOCTYPE REQ SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsV1.00.dtd'>
      <REQ>
      <VER>1.0</VER>
   <ACCOUNT>
       <ID>18244</ID>
   </ACCOUNT>
   <MESSAGE>
       <TAG>hello</TAG>
       <TEXT>Bulk Push SMS-sample test</TEXT>
      <TYPE>0</TYPE>
       <MID>1</MID>
       <SMS FROM='' TO = '919886080768' INDEX = '1' TAG = ''></SMS>
   </MESSAGE>
</REQ>
```

OUTPUT

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'> <RESULT>
<REQUEST-ERROR>
    <ERROR>
          <CODE>102</CODE>
          <DESC>Client 18244 is not registered</DESC>
    < /ERRO
      R>
</REQUEST-ERROR>
</RESULT>
```

2.11.7.3 Message valid, Mobile numbers invalid

In the following example, the text is intended to be sent to 3 recipients, out of which 2 recipients have invalid mobile numbers.

When this happens, the API will send out 1 message for which the number has been specified correctly, and return an error for the other 2 mobile numbers.

INPUT

```
<!DOCTYPE REQ SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsV1.00.dtd'>
<REQ>
   <VER>1.0</VER>
   <ACCOUNT>
       <ID>38244</ID>
   </ACCOUNT>
   <MESSAGE>
       <TAG>hello</TAG>
       <TEXT>text on two lines </TEXT>

      < TYPE>0</TYP E>
       <MID>1</MID>
         <SMS FROM=’’ TO = '919886080768' INDEX = '1' TAG = ''></SMS>
<SMS FROM='MyToday' TO = '+919886080769' INDEX = '2' TAG =
''></SMS>
<SMS FROM='MyToday' TO = '+919886080769' INDEX = '3' TAG =
''></SMS>
   </MESSAGE>
< /REQ >
```

OUTPUT

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID ='835'>
 <MID SUBMITDATE='2008-05-27 17:09:19' ID='1' TAG = 'hello' TID = '121136'>
<ERROR INDEX = '2'>
<ERROR>
<CODE>108</CODE>
 <DESC>Mobile number +919886080769 invalid</DESC>
</ERROR>
 </ERROR>
 <ERROR INDEX = '3'>
<ERROR> <CODE>108</CODE>
 <DESC>Mobile number +919886080769 invalid</DESC>
</ERROR>
 </ERROR>   </MID>
 </RESULT>
```

2.11.7.4 Some messages with Mobile numbers invalid

In the following example, there are 2 messages being sent out to a set of 3 mobile numbers each. The message with ID 2 has 2 invalid numbers in its specification. In this scenario, message to valid numbers go out and the invalid ones don't. In this case, The 2 numbers that failed for the second message, will be mentioned in the error response that the API returns.

INPUT

```
<!DOCTYPE REQ SYSTEM
'http://localhost:8080/BulkSms/BulkSmsV1.00.dtd'>
<REQ>
   <VER>1.0</VER>
   <USER>
        <USERNAME>99670xxxxx</USERNAME>
        <PASSWORD>abcde</PASSWORD>
   </USER>
   <ACCOUNT>
       <ID>38244</ID>
   </ACCOUNT>
<MESSAGE>
       <TAG>helloo</TAG>
       <TEXT>text on one lines </TEXT>
      <TYPE>0</TYPE>
       <MID>1</MID>
       <SMS FROM='' TO = '919886080768' INDEX = '1' TAG = ''></SMS>
<SMS FROM='super' TO = '919886080770' INDEX = '2' TAG = ''></SMS>
<SMS FROM='super' TO = '919886080771' INDEX = '3' TAG = ''></SMS>
   </MESSAGE>
   <MESSAGE>
       <TAG>hello</TAG>
       <TEXT>text on two lines </TEXT>
      <TYPE>0</TYPE>
       <MID>2</MID>
       <SMS FROM='' TO = '919886080772' INDEX = '1' TAG = ''></SMS>
<SMS FROM='MyToday' TO = '+919886080769' INDEX = '2' TAG = ''></SMS>
<SMS FROM='MyToday' TO = '+919886080769' INDEX = '3' TAG = ''></SMS>
   </MESSAGE>
</REQ>
```

OUTPUT

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID ='846'>
    <MID SUBMITDATE='2008-05-27 17:14:04' ID='1' TAG = 'hello' TID =
'121148'>


    </MID>  <MID SUBMITDATE='2008-05-27 17:14:04' ID='2' TAG =
'hello' TID = '121149'>
```

```
            <ERROR INDEX = '2'>



     <ERROR>
     <CODE>108</CODE>
<DESC>Mobile number +919886080769
        invalid</DESC>
     </ERROR>
            </ERROR>
            <ERROR INDEX = '3'>



     <ERROR>
     <CODE>108</CODE>
<DESC>Mobile number +919886080769
        invalid</DESC>
     </ERROR>
            </ERROR>
            </MID>
         </RESULT>
```

2.11.7.5 Some messages invalid

A message may be invalid if for example, the index of the mobile numbers in the specification is duplicated. When such a scenario happens the whole message is considered as invalid and none of the mobile numbers in the message will receive messages. No transaction id will be returned for invalid messages. All other messages, if any in the request, that have the correct specification will be delivered as expected.

In the following example, Message ID 1 is invalid, since the message indices are not unique. However Message ID 2 is a completely valid message with 3 recipients. Message ID 1 will not be sent out completely, however Message ID 2 will be delivered to all the recipients.

INPUT

```
<!DOCTYPE REQ SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsV1.00.dtd'>
   <VER>1.0</VER>
   <ACCOUNT>
       <ID>38244</ID>
   </ACCOUNT>
<MESSAGE>
       <TAG>helloo</TAG>
       <TEXT>text on one lines </TEXT>
      <TYPE>0</TYPE>
       <MID>1</MID>

        <SMS FROM='super' TO = '919886080768' INDEX = '1' TAG =
 ''></SMS>
 <SMS FROM='super' TO = '919886080770' INDEX = '1' TAG = ''></SMS>
 <SMS FROM='super' TO = '919886080771' INDEX = '3' TAG = ''></SMS>
   </MESSAGE>
 <MESSAGE>
       <TAG>hello</TAG>
       <TEXT>text on two lines </TEXT>
      < TYPE>0</TYP
          E>
       <MID>2</MID>
       <SMS FROM=’’ TO = '919886080768' INDEX = '1' TAG = ''></SMS>
 <SMS FROM='super' TO = '919886080770' INDEX = '2' TAG = ''></SMS>
 <SMS FROM='super' TO = '919886080771' INDEX = '3' TAG = ''></SMS>
   </MESSAGE>
 </REQ>
```

OUTPUT

```
<!DOCTYPE RESULT SYSTEM
'http://bulkpush.mytoday.com/BulkSms/BulkSmsRespV1.00.dtd'>
<RESULT REQID ='850'>
    <MID SUBMITDATE='2008-05-27 17:19:36' ID='1' TAG = 'hello'> <
          ERRO
            R>
                 <CODE>110</CODE>
               <DESC>Indexes not unique</DESC>
          < /ERRO R>
    </MID>  <MID SUBMITDATE='2008-05-27 17:19:36' ID='2' TAG =
'hello' TID = '121153'>


    </MID>
</RESULT>
```

2.12 Using HTTPS for APIs

To invoke our APIs using HTTPS, one needs to replace "http://" with "https://" in the URI, or Web address.

When a user connects to a website via HTTPS, the website encrypts the session with a digital certificate.

You can send secured requests to the bulkpush.mytoday.com using HTTPS Connection.

The bulkpush.mytoday.com server listens on port 443 and expects an encrypted request. To make secured HTTPS requests to the bulk-push API, instantiate an open SSL enabled net http client.

Now you can call any of the APIs using https://bulkpush.mytoday.com/BulkSms/

If not already installed, you might have to install the open SSL extension to use HTTPS URLs. If you keep trusted certificates on your computer, you can have Open-SSL verify the server's certificates. If not, your conversation with the server will be confidential, but you won't be able to definitively authenticate the server. It might be an impostor.

On Debian/Linux the CA-Certificates package installs a set of trusted server certificates in the directory /etc/ssl/certs. You can set the object's certificates-authority path to that directory and set the verify mode to verify-peer.

Now Open-SSL can verify that you are actually talking to the web server at bulkpush.mytoday.com and not an impostor.

The SSL certificate for mytoday.com is signed by Digicert.If you already have Digicert certificate installed on your computer, you can verify

bulkpush.mytoday.com's signature. That is "If I trust Digicert, I can trust mytoday.com".

2.13 Methods to Fetch Delivery reports

2.13.1 Reports VIA Bizbond Panel:

You can download the delivery reports from BizBond Panel for the same you need to login to http://biz.mytoday.com/web/ link, then click on Reports tab. You will get to below page

To download details reports use first option “Download Detailed Reports for “ , use select and click on download. Second option ”Search From” can be used for searching delivery status of one mobile number for single day from single channel.

2.13.2 Delivery Reports VIA API

The base URL for Delivery Reports will be: http://stats.mytoday.com/dlr_api

Steps for calling API

You need to call the API:

1) http://stats.mytoday.com/dlr_api?feedid=12345&date=2009-12-05

Feedid is a mandatory parameter for which you must specify a value. Date is one of the optional parameters. If do not specify a date, by default, the delivery report for today's date will be shown. Refer to API Parameters for API Parameters.
If there is any error while specifying the API parameters, error message will be displayed corresponding to the type of error. Refer to 2.4 Error Codes to view the various error codes/ messages. This will initiate the request to fetch Delivery Reports, asynchronously.

When you call the API, the Delivery Report Request Transaction ID will be returned. This will need to be used for further requests to fetch the Delivery Report.

<RESULT>DTXNID</RESULT> DTXNID is the Delivery Report Request Transaction ID. It is alphanumeric.

For consequent requests to fetch the Delivery Report, the API needs to be invoked the DLR Request Transaction ID parameter (dtxnid=<DTXNID>):

2) http://stats.mytoday.com/dlr_api?feedid=12345&date=2009-12-05&dtxnid=6xyjiulsd24afsf

You get the status as- <RESULT>FETCHING<RESULT> Or
<RESULT>DONE</RESULT>

FETCHING implies that the Delivery Report retrieval is in progress. The API needs to be invoked in a loop till status is returned as DONE.

DONE status indicates that the delivery report can be retrieved in the next API call.

Resend the API request, with an additional parameter ack=1:

3) http://stats.mytoday.com/dlr_api?feedid=12345&date=2009-12-05&dtxnid=6xyjiulsd24afsf&ack=1

The Delivery Report will be available in the required format.

2.13.3 Delivery Reports VIA Ping Back

Using this option we can pingback delivery reports to customers in real time. For the same customers need to provide the pingback URL and delivery reports parameters which needs to be pass.

Below is the example of delivery reports Ping Back:

This will be a http GET call to the URL provided by you.

For better scaling and support, these are the few parameters:

Parameters	Field type	Sample Data	Description
reqid	character	3453456723	the id that is originally returned to the client per request.
mobile	integer (upto 64 bytes)	919000000000	the mobile number tat message was sent to
delv_date	datetime	2009-02-20 00:00:00	the date and time when message was delivered.
status	char	S	**Status** will be **S** for submit / **N** for Not-Delivered / **D** for delivered
feedid	integer (upto 64 bytes)	534867	The feedid from which client sent the message

Sample **URL** will be like:
https://www.xyz.com/dlrnotification.jsp?reqid=3453456723

2.13.4 Delivery reports Via FTP upload

This is simple way of delivery reports sharing in which complete days delivery reports will be uploaded on FTP location shared by client

2.14 Please find the delivery status description:-

1) Delivered - Delivered numbers.

2) Absent subscriber - These are the users who are continuously coming in/out Network because of which the message did not get delivered. Here we retry the message for 4 hours from the Gateway and if still it fails then we put him in the same status in reports.

3) Invalid subscriber -

i) These are the users for whom we received a response that this is an Invalid user from the operator when we tried sending the message to them. If required we can give you the confirmed invalid numbers.

ii) There can also be a case when you push the message to 10 digit mobile number, but that number is not Active use. It may also be a case, that the number is with the SIM card seller and it has not been purchased so far.

iii) You have sent a message to a number which is not a valid 10 digit or 12 digit (with '91' as prefix) mobile number.

4) SMS/Message Inbox full - As the name suggests these are the users whose handset inbox was full when the message was tried to be delivered to them.

5) Blacklist - These are blacklisted numbers at the Operator end.

Probable Reasons for blacklisting:

1) The number might have been used for illegal activities in past

2) Nonpayment of bills

3) The number might belong to a V.V.VIP person of an organization

4) The number might belong to an VVIP Government official, minister, etc.

5) Government of India has asked all operators to not send sms to that number.

6) Complaints from other mobile users about that number
7) User might have done many complaints to operator/TRAI regarding unsolicited sms being received by him.

6) NDNC Reject - These are the numbers registered in NCPR database.

7) Undelivered - These are the users for whom message got failed after multiple retries due to various reasons like operator end/user end issues.

8) Submitted to Network - These are the messages for which we haven't received the delivery reports from the Operator. But, it doesn't mean that the message were not Delivered to the end user. As per our observation, 85 to 90% of messages are delivered to the end user, but we haven't received delivery acknowledgement/reports from the Operator. In such scenario, we wait for approximately 24 to 48 hours for the reports to arrive from the Operator end. If the reports are received, those are updated in the Online panel instantly. In case, if the reports are not received, the delivery status will remain as 'Submitted to n/w'.

9) Dropped -

i) These are the numbers belonging to J&K (Jammu and Kashmir) series. There is ban from Govt of India since past few years, because of which sms cannot be delivered on these numbers.

or

ii) Dropped by the operator from their end. Services to the SIM card (mobile number) will be discontinued from the Operator, if there is no usage i.e no voice calls (incoming & outgoing) , SMS and data for any continuous period of 60 days from the date of first call.

10) Expired - These are the numbers to which the messages are not delivered after multiple retires from the Operator end because of the remote operator end issues.

11) International Roaming: These are the users who are in International roaming with Indian numbers at the time when the message was pushed to them.

12) Force Expired - These are the numbers to which messages are dropped due to rules in our system or due to invalid numbers in the uploaded base or the client has requested to drop the messages.

13) Duplicate Message Drop : If the same message content is sent on the same mobile number within 30 min, the duplicate messages will be dropped.

14) Pending - These are the messages which are not sent out of our system.