



Touch Maldives

Corporate SMS Products

API – A Developer's Guide

Document Revision History

Date	Revision	Revised By	Approved By
October 31, 2013	10312013	Mohamed Azeem	
August 19, 2013	08192013	Mohamed Azeem	
June 4, 2013	06042013	Mohamed Azeem	

TABLE OF CONTENTS

Table of Contents	i
1 Introduction	1
2 Send SMS From Application	2
2.1 By HTTPS GET	2
2.2 By XML POST	3
3 Check SMS Delivery Status	5
4 Receive Incoming SMS to Client Application	6

1 INTRODUCTION

2 SEND SMS FROM APPLICATION

2.1 By HTTPS GET

This is the simplest interface that can be implemented by a client application. However this interface is not recommended if the application would like to keep track of SMS delivery status to handsets. Because for successful submissions, the interface returns a textual description of status only, i.e. "OK".

Interface details:

<https://bulkmessage.com.mv/jsp/receiveSMS.jsp?userid=xx&password=xx&to=9607783989&text=testFromDhiraagu>

Parameters:

1. **userid**: account user name.
2. **password**: account password.
3. **to**: mobile number having "960" prefixed.
4. **text**: the message to be sent (URL encoded).

Response:

1. "OK": if submission is successful.
2. "Failed: description", if there was any issue in message submission.

2.2 By XML POST

Interface: <http://bulkmessage.com.mv/partners/xmlMessage.jsp>

XML POST Request:

```
<?xml version="1.0" encoding="UTF-8" ?>
<TELEMESSAGE>
  <TELEMESSAGE_CONTENT>
    <MESSAGE>
      <MESSAGE_INFORMATION>
        <SUBJECT>testing push API</SUBJECT>
      </MESSAGE_INFORMATION>
      <USER_FROM>
        <CIML>
          <NAML>
            <LOGIN_DETAILS>
              <USER_NAME>AccountUserName</USER_NAME>
              <PASSWORD>AccountPassword</PASSWORD>
            </LOGIN_DETAILS>
          </NAML>
        </CIML>
      </USER_FROM>
      <MESSAGE_CONTENT>
        <TEXT_MESSAGE>
          <MESSAGE_INDEX>0</MESSAGE_INDEX>
          <TEXT>sending SMS through XML POST</TEXT>
        </TEXT_MESSAGE>
      </MESSAGE_CONTENT>
      <USER_TO>
        <CIML>
          <DEVICE_INFORMATION>
            <DEVICE_TYPE DEVICE_TYPE="SMS"/>
            <DEVICE_VALUE>7783989</DEVICE_VALUE>
          </DEVICE_INFORMATION>
        </CIML>
      </USER_TO>
    </MESSAGE>
  </TELEMESSAGE_CONTENT>
  <VERSION>1.6</VERSION>
</TELEMESSAGE>
```

The XML POST request shown above contains all the required parameters, more importantly developers should be aware that the SMS delivered would be a concatenated string of SUBJECT and TEXT values. Hence SUBJECT field can accept without a value. If the SMS to be sent is longer than **160** characters, MESSAGE_INDEX can be used to represent the sequence of the SMS, thus the handset can recognise how to display the received SMS text.

The following snapshot illustrates the structure of the XML response:

```
▼<TELEMESSAGE>
  ▼<TELEMESSAGE_CONTENT>
    ▼<RESPONSE>
      <MESSAGE_ID>77338616</MESSAGE_ID>
      <MESSAGE_KEY>877254683516104416555762518966</MESSAGE_KEY>
      <RESPONSE_STATUS>100</RESPONSE_STATUS>
      ▼<RESPONSE_STATUS_DESC>
        Message added successfully. Returned MessageID from db is 77338616
      </RESPONSE_STATUS_DESC>
    </RESPONSE>
  </TELEMESSAGE_CONTENT>
  <VERSION>1</VERSION>
</TELEMESSAGE>
```

The sending application should provide a textual acknowledgement as “OK” for the above response from the gateway.

3 CHECK SMS DELIVERY STATUS

Delivery status check requires message ID and message key, hence if this feature is required, the client application should implement XML POST method for sending messages.

Interface: <http://bulkmessage.com.mv/partners/xmlMessage.jsp>

XML POST Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<TELEMESSAGE>
  <TELEMESSAGE_CONTENT>
    <MESSAGE_STATUS_QUERY>
      <MESSAGE_ID>77339431</MESSAGE_ID>
      <MESSAGE_KEY>933632706492826821420474656539</MESSAGE_KEY>
    </MESSAGE_STATUS_QUERY>
  </TELEMESSAGE_CONTENT>
  <VERSION>1.6</VERSION>
</TELEMESSAGE>
```

XML Response:

```
▼<TELEMESSAGE>
  ▼<TELEMESSAGE_CONTENT>
    ▼<MESSAGE_STATUS>
      <STATUS_ID>2501</STATUS_ID>
      <STATUS_DESCRIPTION>SMS was delivered to handset</STATUS_DESCRIPTION>
      <MESSAGE_ID>77339431</MESSAGE_ID>
    ▼<RECIPIENT_STATUS>
      <RECIPIENT_NAME>960-7-783989</RECIPIENT_NAME>
      ▼<DEVICE>
        <TYPE>60</TYPE>
        <VALUE>960-7-783989</VALUE>
        <STATUS>2501</STATUS>
        <DESCRIPTION>SMS was delivered to handset</DESCRIPTION>
        <STATUS_DATE>20120528 07:39:43</STATUS_DATE>
      </DEVICE>
    </RECIPIENT_STATUS>
  </MESSAGE_STATUS>
</TELEMESSAGE_CONTENT>
<VERSION>1</VERSION>
</TELEMESSAGE>
```

4 RECEIVE INCOMING SMS TO CLIENT APPLICATION

This feature is available only in “Corporate SMS” accounts. In order to receive incoming SMS details to a client application, a dedicated short-code must be attached with the CorporateSMS account.

Clients can configure required settings to define the way incoming SMS is to be forwarded to external application.

1. Login to the account.
2. Go to Options, then Filtering rules.
3. Click on the “New Rule”.
4. Define a Rule Name (e.g. fowardSMSToApplication)
5. Under “Select one or more of the following actions”
 - a. Enter the application ¹URL to “Other” field.
 - b. Set the “Type” as Partner.
6. Leave all other fields unchanged.
7. Click “Add” to create the rule.

The following snapshot illustrates how the client side configuration is done.

Welcome
 Compose
 Folders
 Address Book
 Options
 View Usage
 Filtering Rules
 Sign Off

ADD CANCEL

New Filtering Rule

Rule Name:

Select Account: other (incoming messages to Inbox folder)

If a message matches **ALL** of the conditions below, the action will be activated.
 Note: Empty phrases are ignored

Select Condition:

From: case sensitive

To/Cc: case sensitive

Subject: case sensitive

Select one or more of the following actions:

1. Forward to my: @

Other Type

2. File in:

3. Auto Reply:

ADD CANCEL

¹ URL should accept POST messages.

Once rule is saved, Click “Filtering rules” again to find the rules created, which rule is active, etc. The following snapshot illustrates the saved rules.

The screenshot displays the 'Filtering Rules' management interface. On the left is a navigation menu with links: Welcome, Compose, Folders, Address Book, Options, View Usage, Filtering Rules (highlighted), and Sign Off. The main area is titled 'Filtering Rules' and contains several buttons: NEW RULE, DELETE, ENABLE, and DISABLE. Below these are status indicators: a checked box for 'Enabled', a box with an 'X' for 'Disabled', a box with a red circle for 'Invalid', and icons for 'Edit Rule' and 'Delete Rule'. A status line indicates '1 - 1 of 1 rule'. A table shows the details of the rule:

1. Rule Name: forwardToTest	
Account	other (incoming messages to Inbox folder)
Condition	
Action	Forward message to: http://172.16.9.20:8006/PostReciever/RecievePost.aspx - Partner

Below the table, another status line shows '1 - 1 of 1 rule' and a second set of buttons: NEW RULE, DELETE, ENABLE, and DISABLE.

A successfully configured application should be able to receive incoming SMS. SMS gateway POSTs incoming messages to the client application in an XML message. The XML on the next page represents a sample message, and clients should expect incoming messages to contain the same structure. SMS may be split to multiple messages when the text length is longer than 160 characters. Hence client may require to lookup the “MESSAGE_INDEX” and “ORIGINAL_MESSAGE_ID” properties to realise the entire SMS message.

Note: The receiving application must respond with “OK” as an acknowledgement. E.g. the partner application reads POST message and returns OK.

```

<?xml version="1.0" encoding="UTF-8"?>
<TELEMESSAGE>
  <TELEMESSAGE_CONTENT>
    <MESSAGE>
      <MESSAGE_INFORMATION>
        <LOCALE>
          <LOCALE_LANGUAGE_ID>en</LOCALE_LANGUAGE_ID>
          <LOCALE_COUNTRY_ID>US</LOCALE_COUNTRY_ID>
        </LOCALE>
        <SUBJECT>Re:</SUBJECT>
        <TIME_STAMP>20130604 08:51:11</TIME_STAMP>
        <SCHEDULE_TO>20130604 08:51:11</SCHEDULE_TO>
      </MESSAGE_INFORMATION>
      <USER_FROM>
        <CIML>
          <NAML>
            <LOGIN_DETAILS>
              <USER_NAME>
            </USER_NAME>
            <PASSWORD>
            </PASSWORD>
            </LOGIN_DETAILS>
            <PERSON_DETAILS>
              <FIRST_NAME>960-7-783989</FIRST_NAME>
              <LAST_NAME />
              <TIME_ZONE>GMT+05:00</TIME_ZONE>
              <STATE />
              <COUNTRY_ID />
              <GENDER>0</GENDER>
              <SERVICE_LEVEL>15</SERVICE_LEVEL>
            </PERSON_DETAILS>
          </NAML>
          <DEVICE_INFORMATION>
            <DEVICE_TYPE DEVICE_TYPE="SMS" />
            <DEVICE_VALUE>960-7-783989</DEVICE_VALUE>
          </DEVICE_INFORMATION>
        </CIML>
      </USER_FROM>
      <MESSAGE_CONTENT>
        <TEXT_MESSAGE>
          <MESSAGE_INDEX>0</MESSAGE_INDEX>
          <TEXT>this is really nice. Testing integration with test application</TEXT>
        </TEXT_MESSAGE>
      </MESSAGE_CONTENT>
      <USER_TO>
        <CIML>
          <NAML>
            <LOGIN_DETAILS>
              <USER_NAME>AccountUserName</USER_NAME>
              <PASSWORD>AccountPassword</PASSWORD>
            </LOGIN_DETAILS>
            <PERSON_DETAILS>
              <FIRST_NAME>Mohamed</FIRST_NAME>
              <LAST_NAME />
              <TIME_ZONE>GMT+05:00</TIME_ZONE>
              <STATE />
              <COUNTRY_ID />
              <GENDER>0</GENDER>
              <SERVICE_LEVEL>312</SERVICE_LEVEL>
            </PERSON_DETAILS>
          </NAML>
          <DEVICE_INFORMATION>
            <DEVICE_TYPE DEVICE_TYPE="EmailAddress" />
            <DEVICE_VALUE>test@testEmail.com</DEVICE_VALUE>
          </DEVICE_INFORMATION>
        </CIML>
      </USER_TO>
      <ORIGINAL_MESSAGE_ID>77352193</ORIGINAL_MESSAGE_ID>
    </MESSAGE>
  </TELEMESSAGE_CONTENT>
  <VERSION>1</VERSION>
</TELEMESSAGE>

```