Enterprise SMS

Implementation Guide

This document provides guidelines and examples on how to implement the ESMSlib library and ESMSWS.php

Prepared by

Mobitel (Pvt) Ltd
108, W. A. D. Ramanayake Mawatha,
Colombo 02, Sri Lanka
Tel : +94 (0) 11 2330550
Web : www.mobitel.lk

Document version 1.8 19 February 2018

Table of Contents

1 Introduction

Please follow the guidelines examples below to implement the Enterprise SMS web service and ESMSlib library.

1.1 Prerequisites

- An internet connection
  - ESMSlib requires an internet connection to operate. In case internet is restricted please ensure that your application server is able to communicate with the URL: http://smeapps.mobitel.lk:8585 .
  - Non-Java developers can access the service via a web service located at the URL: http://smeapps.mobitel.lk:8585/EnterpriseSMSV3/EnterpriseSMSWS?wsdl .
- Dependencies
  - Add a web service reference to the url: http://smeapps.mobitel.lk:8585/EnterpriseSMSV3/EnterpriseSMSWS?wsdl to import all the required methods of the web service to your project file.

- Username and password
  - You would have been provided with a username and password. If not please contact your account manager.

2 Testing the service

Once you have added the web service reference to your project you can test the ESMS service by running the following code:

```
lk.mobitel.smeapps.user  user1= new lk.mobitel.smeapps.user();
user1.username = "YOUR_USERNAME";
user1.password = "YOUR_PASSWORD";

lk.mobitel.smeapps.EnterpriseSMSImplService client = new
lk.mobitel.smeapps.EnterpriseSMSImplService();
Console.WriteLine(client.serviceTest(user1));
Console.ReadKey();
```

Please note that only the **username** and **password** properties need be filled

If the application prints “SUCCESS”, it works!!

3 Handling Sessions

This chapter contains details on how to start/stop and query sessions.

3.1 How to Login

Some example login code to create a session for the user:

```
lk.mobitel.smeapps.user  user1= new lk.mobitel.smeapps.user();
user1.username = "YOUR_USERNAME";
user1.password = "YOUR_PASSWORD";

lk.mobitel.smeapps.EnterpriseSMSImplService client = new
lk.mobitel.smeapps.EnterpriseSMSImplService();


lk.mobitel.smeapps.session session1 = new lk.mobitel.smeapps.session();
session1= client.createSession(user1);
```

3.2 Query aSession

Querying a session can be done using the following code:

```
lk.mobitel.smeapps.EnterpriseSMSImplService client = new
lk.mobitel.smeapps.EnterpriseSMSImplService();


lk.mobitel.smeapps.session session1 = new lk.mobitel.smeapps.session();
Boolean isSession= client.isSession(session1);
```

This will return weather the session is active or not.

3.3 Renew aSession

Renewing sessions will be done automatically through the API. However, if you wish to renew a session manually it can be done in the following manner:

```
lk.mobitel.smeapps.EnterpriseSMSImplService client = new
lk.mobitel.smeapps.EnterpriseSMSImplService();


lk.mobitel.smeapps.session session1 = new lk.mobitel.smeapps.session();
client.renewSession(session1);
```

Each renewal extends the expiry date of the session by 24hours.

3.4 Logout from a Session

It is highly recommended that you log out of the ESMS application on exiting your system. This ensures security and session consistency. Logging out from a session can be done in the following manner:

```
lk.mobitel.smeapps.EnterpriseSMSImplService client = new
lk.mobitel.smeapps.EnterpriseSMSImplService();

client.closeSession(session1);
```

Here session1 is the created session earlier.

4 Sending SMSs

There are several methods of sending SMS. They can be sent to a list of numbers or a group.

4.1 Send Standard SMSs

Sending a normal SMS can be done in the following manner:

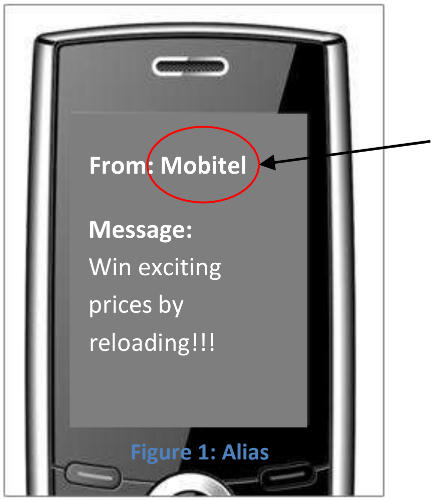
```
lk.mobitel.smeapps.smsMessage msg = new lk.mobitel.smeapps.smsMessage();
msg.message = "This is test";
msg.sender = “ALIAS”;

// Create list of recipients
 var myList = new List<string>();

// Add items to the list
myList.Add("94716980189");
myList.Add("94710793250");

msg.recipients=myList.ToArray();
msg.messageType=1;

Console.WriteLine(client.sendMessages(session1, msg));
```

According to the above code four properties of the smsMessage object are filled.

- message => Message to be sent as string
- recipients => An array of the message receivers phone numbers
- sender => Alias or Sender Mask
- messageType => For Promotional message type the value is 1 whereas 0 if for normal messages type

The ***Alias*** is what appears as the sender address of the SMS as shown in Figure 1.

Finally sendMessage() method is called with the created session object and the message object as arguments as shown in above code.



From: Mobitel

Message:
Win exciting
prices by
reloading!!!

Figure 1: Alias

Alias.setAlias("Mobitel")

The response codes are as follows:

- **200** - Message received OK
- **151** - invalid session
- **152** – session is still in use for previous request
- **155** - service halted
- **156** - other network messaging disabled
- **157** - IDD messages disabled
- **159** - failed credit check
- **160** - no message found
- **161** - message exceeding 160 characters
- **162** – invalid message type found
- **164** - invalid group
- **165** - no recipients found
- **166** - recipient list exceeding allowed limit

- **167** - invalid long number
- **168** - invalid short code
- **169** - invalid alias
- **170** - black listed numbers in number list
- **171** - non-white listed numbers in number list
- **175** - deprecated method
- **200** - message sent OK

4.2 Send Multi-Language SMSs

This method can be used send messages with any especial character including English and Sinhala characters. The method is same as the previous method except **smsMessageMultiLang()** object is created in sending multilanguage messages.

```
lk.mobitel.smeapps.smsMessageMultiLang msg = new lk.mobitel.smeapps. smsMessageMultiLang ();
msg.message = "This is test";
msg.sender = “ALIAS”;

// Create list of recipients
 var myList = new List<string>();

// Add items to the list
myList.Add("94716980189");
myList.Add("94710793250");

msg.recipients=myList.ToArray();
msg.messageType=1;

Console.WriteLine(client.sendMessagesMultiLang(session1, msg));
```

5 Receiving SMSs

5.1 Getting SMSs from a Short Code

```
lk.mobitel.smeapps.smsMessage[]  longCodeMsg=
client.getMessagesFromShortcode(session1, “55599”);
```

The ***getMessagesFromShortcode()*** returns a list of type SmsMessagewhich can be used to process the received messages.

5.2 Getting SMSs from a Long Number

```
lk.mobitel.smeapps.smsMessage[] longCodeMsg =
client.getMessagesFromLongNUmber(session1, “0719889988”);
```

The ***getMessagesFromLongcode()*** returns a list of type SmsMessage which can be used to process the received messages.

5 Recommended Coding Practices

- If you keep receiving a 152 code try waiting for between 100milliseconds – 1 second before trying again.
- Avoid using special characters such as {, [,], }, ^,\, |, ~ in the message content which will reduce the allowed character count from 160 to smaller value due to international encoding standards.
- Use number format as 947XXXXXXX for mobile numbers and 94XXXXXXXX for all local numbers to avoid fixed-line numbers are considered as IDD numbers.