Enterprise SMS

Implementation Guide

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

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 July 2018

Table of Contents

1 Introduction 3
1.1 Prerequisites 3
2 Testing the service 3
3 Handling Sessions 4
3.1 How to Login 4
3.2 Query a Session 4
3.3 Renew a Session 4
3.4 Logout from a Session 5
3.5 Exit ESMS API 5
4 Sending SMSs .....
4.1 Send Standard SMSs 5
4.2 Send Multi-Language SMSs 7
5 Receiving SMSs .....
5.1 Getting SMSs from a Short Code 7
5.2 Getting SMSs from a Long Number 8
6 Recommended Coding Practices 9

1 Introduction

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

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 .
  - can access the service via a web service located at the URL: http://smeapps.mobitel.lk:8585/EnterpriseSMSV3/EnterpriseSMSWS?wsdl .
- Dependencies
  - If you are developing using PHP make sure that Apache server with PHP is installed and the **ESMSWS.php** is included in your necessary PHP scripts.
- Username and password
  - You would have been provided with a username and password. If not please contact your account manager.

2 Testing the service

Web service users can use the "***testService()***" method and which requires following parameters accordingly:

{id, username, password, customer}

Please note that only the **username** and **password** properties need be filled whereas **id** and **customer** properties are left empty as shown below.

```
serviceTest("","username","password","");
```

3 Handling Sessions

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

3.1 How to Login

Some example login code:

Use ***createSession()*** method which requires the following properties to create a new session in the PHP file:

{id, username, password, customer}

Please note that only the **username** and **password** properties need be filled whereas **id** and **customer** properties are left empty as shown below.

```
$session=createSession('',$username,$password,'');
```

The $session variable can be used in the methods to send and receive SMS as described in the following chapters.

The return object of type session will contain the following properties:

{id, sessionId, user, expiryDate, isActive}

3.2 Query aSession

Querying a session can be done using the following code:

```
$session_validity=isSession($session);
```

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:

```
renewSession($session);
```

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 using "***closeSession()***" method and pass an object of type **Session** with the following properties:

{id, sessionId, user, expiryDate, isActive}

Typically the **Session** object returned from the *createSession()* method should be given as the input.

```
closeSession($session);
```

where $session is the session variable created when creating a session.

.

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 using ***sendMessages()*** method which requires the following properties

{session, alias, message_body, array of reciepients, messageType}

You should create the ***$session*** variable as described previously and pass that variable in to this method as first argument.

- The ***Alias*** is what appears as the sender address of the SMS as shown in Figure 1.
- The ***$message_body*** variable contains the message you want to send as a string
- ***array of reciepients*** is the list of receivers of the message which is an array where numbers can be added up to a total of 500 .
- Final argument is the **message Type** which is 1 for Promotional message type and 0 for normal messages

```
sendMessages($session,$alias,$message_body,array('9471XXXXXXX'), 1);
```

If you are executing multiple consecutive requests, then it is wise to keep the session variable and close the session after all of them are complete. Or else you can close the session right away after you complete the current operation. PHP code for closing the s1ession is described in the previous chapter.

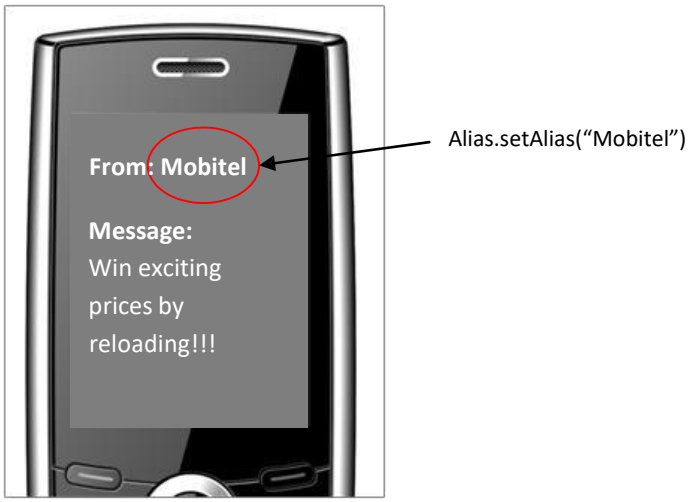

Figure 1: Alias

The response codes of sendMessage() method 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

4.1 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.

{session, alias, message_body, array of reciepients, messageType}

```
sendMessagesMultiLang ($session,$alias,$message_body,array('9471XXXXXXX'), 1);
```

5 Receiving SMSs

5.1 Getting SMSs from a Short Code

```
$received_messages=getMessagesFromShortCode($session,"shortcode");
```

This method will return an array of smsMessageobjects which will have the following attributes:

- message :String
- sender : Object of type 'alias' with property 'alias' of type String
- type: datetime

5.2 Getting SMSs from a Long Number

Web service users can use the "***getMessagesFrom LongNUmber ()***" method and pass objects of type
Session and **LongNumber**.

```
$received_messages=getMessagesFromLongNumber($session,"071XXXXXXX");
```

This method will return an array of smsMessageobjects which will have the following attributes:

- message :String
- sender : Object of type 'alias' with property 'alias' of type String
- type: datetime

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 number