

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 1.6

Tuesday, 12 March 2013

Table of Contents

1	Introduction	3
1.1	Prerequisites	3
2	Testing the service	3
3	Handling Sessions.....	5
3.1	How to Login	5
3.2	Logout from a Session.....	6
3.3	Exit ESMS API	6
4	Sending SMSs	7
4.1	Send Standard SMSs	7
4.2	Getting SMSs from a Short Code	9
4.3	Getting SMSs from a Long Number	10
5	Recommended Coding Practices	11

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/EnterpriseSMS/EnterpriseSMSWS?wsdl> .
- Dependencies
 - This library depends on the **webservices-rt.jar** library.
 - If this application is running on a Glassfish application server, there will be no 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

JAVA

Once you import the ESMSlib library to you project you can test the ESMS service by running the following code:

```
lk.mobitel.esms.User user = new lk.mobitel.esms.User ();
user.setUsername("Your_username");
user.setPassword("Your_Password");

lk.mobitel.esms.test.ServiceTest st = new
lk.mobitel.esms.test.ServiceTest ();
System.out.println(st.testService(u));
```

If the application prints "Welcome TestUser", it works!!

PHP

Create a new PHP file and use the code below to test the service

```
echo serviceTest('', 'Your_username', 'Your_password', '');
```

Call the PHP file from a browser or via the command line and it will print a welcome message.

Web Service:

Web service users can use the “***testService()***” method and pass an object of type **User** with the following properties:

```
{id, username, password, customer}
```

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

3 Handling Sessions

This chapter contains details on how to start/stop and query sessions. All session activities are handled by the `lk.mobitel.esms.session.SessionManager` class (JAVA).

3.1 How to Login

Some example login code:

JAVA

```
User user = new User();
user.setUsername("TestUser");
user.setPassword("Password");

SessionManager sm = SessionManager.getInstance();
sm.login(user);
```

Once this set is done, all session management including session renewal will be done through the `SessionManager` class.

PHP

Use the following code to create a new session in the PHP file:

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

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

Web Service:

Web service users can use the “***createSession()***” method and pass an object of type **User** with the following properties:

```
{id, username, password, customer}
```

Please note that only the **username** and **password** properties need be filled. The return object of type session will contain the following properties:

```
{id, sessionId, user, expiryDate, isActive}
```

Note: the **isActive property is has been deprecated.*

3.2 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:

JAVA

```
SessionManager sm = SessionManager.getInstance();  
sm.logout();
```

PHP

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

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

Web Service:

Web service users can use the “***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.

Note: the **isActive property is has been deprecated.*

3.3 Exit ESMS API

Exiting the ESMSlib API performs a garbage cleanup of the application. It cancels all internal timers and exits the API safely.

JAVA

```
SessionManager sm = SessionManager.getInstance();  
sm.lexit();
```

PHP

This is not required when using PHP.

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:

JAVA

```
Alias alias = new Alias();
alias.setAlias("TEST");

SmsMessage msg = new SmsMessage();
msg.setMessage("This is a test");
msg.setSender(alias);

/** recipients per SMSMessage is limited to 500 */
msg.getRecipients().add("0094711234567");
msg.getRecipients().add("712345678");

try {

    SMSManager smsMgr = new SMSManager();
    smsMgr.sendMessage(msg);

} catch (NullSessionException ex) {

    Logger.getLogger(TestEsms.class.getName()).log(Level.SEVERE
, null, ex);

}
```

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

PHP

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

You should create the `$session` variable as described previously and pass that variable in to this method. 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 session is described in the previous chapter.

Web Service:

Web service users can use the “**sendMessage ()**” method and pass objects of type **Session** and **SmsMessage**. The Session object has the following properties:

{message, messageId, recipients[], retries, sender, sequenceNum, status, time, type, user}

The property **recipients** is a list (or array) to which multiple numbers can be added up to a total of 500.

The property **sender** is an object of Type **Alias** with the following properties:

{id, alias, customer}

Only the **Alias** property need be filled. The property **user** is of type user and can be left as **null**.

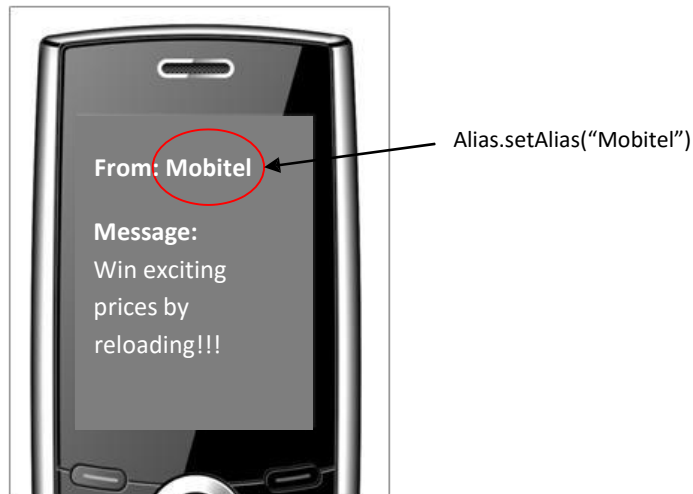


Figure 1: Alias

The response codes are as follows:

- **0** - 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
- **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 Getting SMSs from a Short Code

JAVA

```
Shortcode shortcode = new Shortcode();
shortcode.setShortcode("55599");
SMSManager smsMgr = new SMSManager();

try {

    List<SmsMessage> smsList =
        smsMgr.getMessagesFromShortcode(shortcode);
    processReceivedMessages(smsList);

} catch (NullSessionException ex) {

    Logger.getLogger(TestEsms.class.getName()).log(Level.SEVERE,
        null, ex);

}
```

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

PHP

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

This method will return an array of `smsMessage` objects which will have the following attributes:

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

Web Service:

Web service users can use the “***getMessagesFromShortcode ()***” method and pass objects of type **Session** and **Shortcode**. The **Shortcode** object has the following properties:

{id, Shortcode, customer}

Only the **shortcode** property need be filled. The return value is a list (or array) of type **SmsMessage**.

4.3 Getting SMSs from a Long Number

JAVA

```
LongNumber ln = new LongNumber();
ln.setLongNumber("0719889988");

SMSManager smsMgr = new SMSManager();
try {

    List<SmsMessage> smsList =
        smsMgr.getMessagesFromLongNumber(ln);
    processReceivedMessages(smsList);

} catch (NullSessionException ex) {

    Logger.getLogger(TestEsms.class.getName()).log(Level.SEVERE,
        null, ex);

}
```

PHP

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

This method will return an array of **smsMessage** objects which will have the following attributes:

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

Web Service:

Web service users can use the “***getMessagesFrom LongNumber ()***” method and pass objects of type **Session** and **LongNumber**. The **LongNumber** object has the following properties:

{id, longNumber, customer}

Only the **longNumber** property need be filled. The return value is a list (or array) of type **SmsMessage**.

5 Recommended Coding Practices

- The `SessionManager` is designed to renew a session every 24 hours.
- 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 94XXXXXXX for CDMA numbers to avoid CDMA numbers are considered as IDD numbers.