MessageMedia

Java SMS API

Release 2.0

User Manual

This document outlines installation requirements, methods
and properties for using the MessageMedia Java SMS API.

Last updated

6 June 2012

Contents

Product Information

Product Name	MessageMedia Java SMS API Release 2.0
Purpose	This document lays out the requirements and specifications for the MessageMedia Java SMS API.
Prepared by	Matthew Kwan
	Message4U Pty Ltd trading as MessageMedia

Modifications	Created	16 April 2004
Release	1.0.0.0	1 March 2004
	2.0.0.0	1 March 2004
	2.0.0.1	6 June 2012

Introduction

MessageMedia **Java SMS API**

MessageMedia Java SMS API was created to provide an interface to the MessageMedia SMS messaging server. It has been created using Java allowing MessageMedia Java SMS API to be compatible with all platforms with Java installed.

A key goal in the design of MessageMedia Java SMS API has been to develop a highly user-friendly product which is capable of providing a complete business solution. Message4U Java SMS API

has been implemented with the capabilities to offer an encrypted SSL link to the server.

Packages

MessageMedia Java SMS API can be utilised in one of two ways:

1. smsapi

   The smsapi package provides an interface to the MessageMedia SMS messaging server.

2. securesmsapi

   The securesmsapi package provides the same interface to the MessageMedia SMS messaging server except it connects to the server using an encrypted SSL link.

Although there are two different packages offered for the MessageMedia Java SMS API they are used identically. The securesmsapi package contains more features but these features have been inbuilt and require the same knowledge to implement as the smsapi package does.

Requirements

To be able to use the MessageMedia Java SMS API you will need to have the following:

Internet connection (it uses HTTP port 80)

Java installed

Constraints   The MessageMedia Java SMS API securesmsapi package requires that the version of Java installed on the system is Java 1.4 or later.

Older systems may need to use the smsapi package.

Using the MessageMedia Java SMS API

Installation

To install MessageMedia Java SMS API simply unzip the installation package into the appropriate directory. If you have any problems please don't hesitate to contact us.

Examples

Specific examples of the MessageMedia Java SMS API are available upon request. Please contact us to request one.

Also included in the installation package is a file called "ApiTester.java" which contains example code.

Language Reference

Architecture

The MessageMedia Java SMS API smsapi package has seven classes while the securesmsapi has nine.

smsapi Package

MessageStatus – Class

This class defines constants used to specify a message delivery status.

SmsInterface – Class

This is the main class used to interface with the MessageMedia SMS messaging server.

SmsMessage – Class

This class represents an SMS message.

SmsMessageList – Class

This class represents a list of SMS messages.

SmsReply – Class

This class represents an SMS reply

SmsReplyList – Class

This class represents a list of SMS replies

ValidityPeriod – Class

This class defines constants used to specify the validity period of messages injected into the SMS network.

securesmsapi Package

The securesmsapi contains all the same classes as the smsapi package but with two additional support classes that are not accessed via the API.

X509TrustAllManager – Class

This class is used to create an encrypted SSL link connection to the MessageMedia SMS messaging server.

AllHostnameVerifier – Class

This class is used to create an encrypted SSL link connection to the MessageMedia SMS messaging server.

Constant Values

This section outlines the constant values specified throughout all the classes including their name, type, value and a brief description.

Class: MessageStatus

Name

DELIVERED

Type

short

Value

2

Description

This variable signifies that the message has been successfully delivered.

Name

FAILED

Type

short

Value

3

Description

This variable signifies that the message has not been successfully delivered.

Name

NONE

Type

short

Value

0

Description

This variable signifies that this not a delivery report.

Constant Values

Name

PENDING

Type

short

Value

1

Description

This variable signifies that the message has been accepted by the
network and delivery is pending.

Class: ValidityPeriod

Name

DEFAULT

Type

short

Value

168

Description

This variable signifies that the message is valid for two days.

Name

MAXIMUM

Type

short

Value

255

Description

This variable signifies that the message is valid for sixty three weeks.

Constant Values

Constant Values

Name

MINIMUM

Type

short

Value

0

Description

This variable signifies that the message is valid for five minutes.

Name

ONE_DAY

Type

short

Value

167

Description

This variable signifies that the message is valid for one day.

Name

ONE_HOUR

Type

short

Value

11

Description

This variable signifies that the message is valid for one hour.

Name

ONE_WEEK

Type

short

Value

173

Description

This variable signifies that the message is valid for one week.

Constant Values

Name

 SIX_HOURS

Type

 short

Value

71

Description

This variable signifies that the message is valid for six hours.

Name

 THREE_DAYS

Type

short

Value

169

Description

This variable signifies that the message is valid for three days.

Method Summaries

This section outlines the classes and the methods associated with these classes.

Class: SmsMessage

This class represents an SMS message

Name

SmsMessage()

Parameter

java.lang.String – phoneNumber

java.lang.String – message

long – messageID

int – delay

short – validityPeriod

boolean – deliveryReport

Return Value

None

Description

This is the constructor for creating an SmsMessage. It initializes all the variables inside the SmsMessage class. The delay is in seconds. ValidityPeriod values should be taken from the class with the same name and when the deliveryReport is set to true a report will be requested from the receiving phone. The messageID is used so that a sent message can be linked to a received message.

Name

getDelay()

Parameter

None

Return Value

int

Description

This method returns the delay before sending the message.

Method Summaries

Name

getDeliveryReportRequest()

Parameter

None

Return Value

Boolean

Description

This method returns true if a delivery report has been requested and false if not.

Name

getMessage()

Parameter

None

Return Value

java.lang.String

Description

This method returns the text of the message.

Name

getMessageID()

Parameter

None

Return Value

long

Description

This method returns the message ID.

Method Summaries

Name

getPhoneNumer()

Parameter

None

Return Value

java.lang.String

Description

This method returns the phone number the message is being sent to.

Name

getValidityPeriod()

Parameter

None

Return Value

Short

Description

This method returns how long the message is valid for.

Class: SmsReply

This class represents an SMS reply

Name

getDeliveryStatus()

Parameter

None

Return Value

Short

Description

This method returns the current delivery status which has it possible values described in the MessageStatus class.

Method Summaries

Name

getMessage()

Parameter None

Return Value

java.lang.String

Description

This method returns the contents of the reply.

Name

getMessageID()

Parameter

None

Return Value

Long

Description

This method returns the message ID of the message that it is replying to.

Name

getPhoneNumber()

Parameter

None

Return Value

java.lang.String

Description

This method returns the phone number that sent the reply.

Name

getWhen()

Parameter

None

Return Value

long

Description

This method returns how long ago, in seconds, the reply was received.

Method Summaries

Class: SmsReplyList

This class represents a list of SMS replies.

Name

clear()

Paramete

None

Return Value

None

Description

This method clears all replies message stored in this class.

Name

getReply()

Parameter

Int – n

Return Value

None

Description

This method returns the n'th SmsMessage in the reply list or it throws an IndexOutOfBoundException.

Name

getSize()

Parameter

None

Return Value

int

Description

This method returns the number of replies in the list.

Method Summaries

Class: SmsInterface

This is the main class used to interface with the MessageMedia SMS messaging server.

Name

addMessage()

Parameter

SmsMessage – sm

Return Value

None

Description

This method adds a message to be sent.

Name

addMessage()

Parameter

java.lang.String – recipientNumber

java.lang.String – messageText

long – messageID

int – delay

boolean – deliveryReport

Return Value

None

Description

This method takes all the parameters and creates an SmsMessage and adds the message to be sent.

Method Summaries

Name

changePassword()

Parameter

java.lang.String – newPassword

Return Value

Boolean

Description

This method changes the password on the local machine and server only if the current password has been validated with a connect call. It returns true if the password has been successfully changed.

Name

checkReplies()

Parameter

None

Return Value

SmsReplyList

Description

This method returns the list of replies we have received only if the current password has been validated with a call to the connect method.

*NOTE: Do not run checkReplies() more than once every 60 seconds, this is interface does not support more frequent calls. Running checkReplies() more frequently may result in your account being disabled.

If you require real time delivery of replies please contact your MessageMedia Account Manager or Support and request the documentation for MessageMedia's "Web Push".

Name

clearMessages()

Parameter

None

Return Value

None

Description

This method clears all the messages from the list.

Method Summaries

Name

connect()

Parameter

java.lang.String – username

java.lang.String – password

Return Value

boolean

Description

This method connects to the MessageMedia server using the username and password supplied and makes sure it is responding correctly. It returns true if the connection was successful.

Name

getCreditsRemaining()

Parameter

None

Return Value

Int

Description

This method returns the credit remaining if the connect method has been called. It returns daily credit remaining for post-paid users.

Error Codes

-2 signifies that there was an error while accessing the data.

Method Summaries

Name

getResponseCode()

Parameter

None

Return Value

Int

Description

This method returns the response code received from calls to the methods changePassword, getCreditsRemaining, sendMessages and checkReplies.

Error Codes

100 – OK

400 – Server error

500 – API communications error

510 – Incorrect password

520 – Insufficient daily credit

540 – Insufficient credit

-1 – No response

Name

getResponseMessage()

Parameter

None

Return Value

java.lang.String

Description

This method returns the message that was returned with the response code. This will be null if the value of the getResponseCode is -1.

Method Summaries

Name

sendMessages()

Parameter

None

Return Value

Boolean

Description

This method sends all the messages that have been added with the addMessage method only if a call to the connect method is successful. It returns true if the messages were sent and false otherwise.

Name

setHttpProxy()

Parameter

java.lang.String – proxyHost

int - proxyPort

java.lang.String – username

java.lang.String – password

Return Value

None

Description

Specify a proxy server to be used for HTTP communications. This function should be called before connect. If no username or password is required please assign these fields to be null.

Troubleshooting

Contact Us: Feedback is always welcome. If anything in the document is unclear, or is not working as described, please do not hesitate to contact us:

AUSTRALIA

Tel: 1800 155 228

Email: support@messagemedia.com.au

NZ

Tel: 0800 686 964

Email: support@message-media.co.nz

USA

Tel: 866-751-8337

Email: support@message-media.com

UK

Tel: 0808 234 4874

Email: support@messagemedia.co.uk