

The Hexadimensional Integrator Web Services

Developer's Guide and Reference



Revision History

Summary

Revision	Revision History Summary	Release Date
2.0	Update variable names	01-Oct-2011
1.3	The following new methods have been added: 1.getInboxMessageSingle 2.getInboxMessageMulti 3.getReceiptSingle 4.getReceiptMulti	21-Sept-2011
1.2	1.The variable 'sendernameisalpha' is not being used 2.A new method 'getAccountInfo' has been added	15-June-2011
1.1		
1.0		

Changes in Version 2.0

Several names of member fields and their types have been altered for consistency.

Changes in Version 1.3

•4 new methods have been added to the webservice, which would serve the purpose of retrieving the details of the inbox messages and the receipts (single/multiple), the details of which can be referred to the respective sections below.

(a)getInboxMessageSingle

(b)getInboxMessageMulti

(c)getReceiptSingle

(d)getReceiptMulti

Changes in Version 1.2

•In previous versions, the parameter sendernameisalpha within message-sending functions was used to signify whether the sendername specified was an alphanumeric name (true) or shortcode (false). From this version onwards, this parameter has been deprecated, and its value is no longer being used by the web service.

The parameter itself has been retained, since a change to the parameter would require existing customer stubs and calls to be altered.

This parameter will be eliminated in future versions of the Web Service.

•A new method 'getAccountInfo' has been added to the webservice, the details of which can be referred to the respective sections below.

Contents

Revision History.....	2
Summary.....	2
Changes in Version 2.0.....	2
Changes in Version 1.3.....	2
Changes in Version 1.2.....	2
Introduction.....	5
Integrator.....	5
Web Services.....	5
The Integrator Services.....	5
About this document	5
More about Web Services.....	6
Axis.....	6
WSDL.....	6
WSDL2Java and Creating Stubs.....	6
Hexadimensional Integrator Web Services.....	8
Features of the Integrator Web Services.....	9
Authentication.....	9
Types of Methods.....	9
Methods available.....	11
sendSMS.....	11
sendSMS_String.....	12
sendMultiSMS.....	13
sendMultiSMS_String.....	14
getAccountInfo.....	17
getInboxMessageSingle.....	18
getInboxMessageMulti.....	19
getReceiptSingle.....	20
getReceiptMulti.....	21
Methods Available for testing.....	22
testConnection.....	22
testAuthentication.....	22

Introduction

The Hexadimensional SMS Gateway is a corporate service that allows users to send SMS messages using internet and web-based technologies. This service enables businesses and corporations to easily transmit SMS messages to a huge mass of recipients conveniently and quickly.

Integrator

The Integrator is a set of APIs that allow software developers to integrate the functionality of the SMS Gateway into their applications, for example, send SMS on internal application events.

The Integrator set of APIs comprise:

- The Web-Services API
- The HTTP API

This document details the Web-Services API, and its various functions.

Web Services

Web Services are a generic technology designed to support machines interacting over a network. A Web Service can be accessed over a network and executed on a remote system hosting the requested services. The term 'web services' refers to the client and servers that communicate over the HTTP protocol over the Internet.

The Integrator Services

The Integrator Web Services are a set of Web Services that have been designed for users of Hexadimensional's SMS System to allow their applications to programmatically invoke the server, so as to send SMSs to a large number of recipients conveniently and quickly.

About this document

This document outlines the various functions available to send messages and related information using Web Services. Each function also describes the different parameters returned by it, with examples.

More about Web Services

Web Services are internally executed using XML documents, which are conveyed between the client and the server through one or more HTTP connections.

While Web Services can be invoked in a raw fashion, with direct control of the XML by the application developer, the preferred methodology is to use a set of third-party libraries to send and receive these web service requests and responses. One such free and open-source implementation is the Java-based Apache Axis.

In this document, any references to an actual Web Service library or implementation is taken to be Apache Axis. Though, it should be noted that the use of Web Services, and the functions present in this document and related service, are not limited to Apache Axis, and in fact are present in most current development languages and environments, including Microsoft .NET , PHP, and others.

Axis

SOAP is an XML-based communication protocol and encoding format for inter-application communication. Axis is a framework for constructing SOAP processors such as clients, servers and gateways. It also includes a server which plugs into servlet engines such as Tomcat, extensive support for the *Web Service Description Language (WSDL)* and an emitter tooling that generates Java classes from WSDL.

WSDL

When a service is made available using Axis, there is typically a unique URL associated with that service. If the service URL is accessed in a browser, a message indicating that the endpoint is a web service is displayed. However, on adding "?wsdl" to the end of the URL, a service description for the deployed service is returned as XML to the browser. This WSDL defines the service, and its contained features.

WSDL2Java and Creating Stubs

The Apache Axis implementation includes a utility to generate stubs which can be called by the application, as an intermediary to calls to the web service.

The stubs are created using the following commandline:

```
java org.apache.axis.wsdl.WSDL2Java <url to wsdl>
```

where the URL to the WDSL can be a file path, or an HTTP path, with appropriate prefixes, if required.

On execution of this command, the stubs will be generated in the current directory.

Note that since WSDL2Java is a component of Apache Axis, Axis has to be properly installed and configured for the command to be successful. Installation of Apache Axis is beyond the scope of this document.

Hexadimensional Integrator Web Services

Hexadimensional provides its web services on their platform, the home page of which is accessible at:

<http://msgw.hexadimensional.net>

The web service's service endpoint URL is:

`http://msgw.hexadimensional.net/services/WSIntegrator_1`

and its WSDL is available at:

`http://msgw.hexadimensional.net/services/WSIntegrator_1?wsdl`

If Apache Axis has been properly installed and configured on the client system, the stubs can be created using the following command:

```
java org.apache.axis.wsdl.WSDL2Java \  
http://msgw.hexadimensional.net/services/WSIntegrator_1?wsdl
```

Note: The commandline above should be entered as a single line, without the '\ ' delimiter. The '\ ' delimiter is used on Unix and Linux systems to denote a single-line command that spans multiple lines. Such a delimiter may not be present on your operating system.

Features of the Integrator Web Services

In later sections are detailed the methods contained in the web service. These methods can be invoked by your application to perform the required functionality.

Authentication

The security of your account is maintained by password-based authentication. Almost all of the functions require a username and password, which identifies you as the authorized user of the account.

The username and password can be that of any of the 'Logins' present in your account on the web site. It is recommended that a separate login be setup within your account, for the purpose of Web Services. Creating a separate login also helps while extracting reports of web service usage vs web-based usage by personnel in your organization.

Types of Methods

The list of methods in the later section include several methods that seem to perform the same operation. These have been provided for the following reasons:

String return vs object return

The '_String' functions (for example, 'sendSMS' vs 'sendSMS_String') return all values as a single string object. This feature is included due to systems that can connect to web services, but are not 100% compatible, ie, they are unable to breakdown returned objects. The '_String' versions of these functions can be used, and the components of the string split up using simple string functions.

The components inside such a string are returned as comma-delimited values. Such values can even be saved into a '.CSV' file, to be viewed using any CSV-compatible viewer, such as Microsoft Excel.

Single Vs Multi

Separate functions have been provided to allow multiple-object operations (for example, sendSMS vs sendMultiSMS). Use of these functions minimizes the number of http requests and responses, thereby improving the overall efficiency of the service. Use of these multi functions, where possible, is highly recommended.

Other Notes

Message concatenation (joining of two or more messages) is transparent using the Integrator Web Services. The user can receive multiple messages as one provided they were sent as one message. The OutMsgID, which is the unique ID generated by the server is the same for all parts of the message.

Methods available

sendSMS

This function posts a single SMS to the server. The parameters to be passed to this function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.
String	sendername	The name or shortcode that should appear as the sender of the SMS. This name or shortcode should be approved for your account.
boolean	sendernameisalpha	Pass true if the sendername parameter is an alphanumeric name; false if it is a shortcode.
boolean	requestreceipt	If a delivery report is required for the sent message.
boolean	isflash	If the SMS is intended to be a flash message.
Date	scheduledate	The intended date and time of sending the SMS. Null for immediate sending.
String	recipientname	The name of the recipient. Optional. This field is useful if names of recipients are to be stored in your Sent Messages.
String	recipientMSISDNwithCC	The phone number of the recipient (with country code). The phone number should not include any leading '0's or '+' symbols.
String	messagetext	The text content of the SMS.

It returns an object with the following parameters.

Parameter Name	Description
SendSuccess	True if the sms was posted to the queue successfully; false otherwise. Note that a successful post into the outbound queue on the gateway does not mean or guarantee a successful send or delivery of the message.
ErrorMessage	Error message if any occurred while posting the message to the queue.
OutMessageID	Server generated unique ID representing an SMS successfully posted to the server. It is recommended that you store this value for future reference.
RawSMSCount	Number of parts into which the message was split before posting. Each part is billed separately.

sendSMS_String

This function is similar to the one above. The parameters to be passed to this function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.
String	sendername	The name or shortcode that should appear as the sender of the SMS. This name or shortcode should be approved for your account.
boolean	sendernameisalpha	Pass true if the sendername parameter is an alphanumeric name; false if it is a shortcode.
boolean	requestreceipt	If a delivery report is required for the sent message.
boolean	isflash	If the SMS is intended to be a flash message.
Date	scheduledate	The intended date and time of sending the SMS. Null for immediate sending.
String	recipientname	The name of the recipient. Optional. This field is useful if names of recipients are to be stored in your Sent Messages.
String	recipientMSISDNwithCC	The phone number of the recipient (with country code). The phone number should not include any leading '0's or '+' symbols.
String	messagetext	The text content of the SMS.

It returns a String that is generated from the parameters of an object. The parameters and the string are as follows.

Parameter Name	Description
SendSuccess	True if the sms was posted to the queue successfully; false otherwise. Note that a successful post into the outbound queue on the gateway does not mean or guarantee a successful send or delivery of the message.
ErrorMessage	Error message if any occurred while posting the message to the queue.
OutMessageID	Server generated unique ID representing an SMS successfully posted to the server. It is recommended that you store this value for future reference.
RawSMSCount	Number of parts into which the message was split before posting. Each part is billed separately.

Example sendSMS_String Result:

"T","Success","6458960","1"

sendMultiSMS

This function posts multiple SMSs to the server. The parameters passed to this function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.
String	batchname	Any name to identify the batch of SMS. This name is used purely for your reference, and will appear in your Sent Messages.
String	sendername	The name or shortcode that should appear as the sender of the SMS. This name or shortcode should be approved for your account. This value is used for all of the SMSs to be sent in this request.
boolean	sendernameisalpha	Pass true if the sendername parameter is an alphanumeric name; false if it is a shortcode. This value is used for all of the SMSs to be sent in this request.
boolean	requestreceipt	If a delivery report is required for the sent message. This value is used for all of the SMSs to be sent in this request.
boolean	isflash	If the SMS is intended to be a flash message. This value is used for all of the SMSs to be sent in this request.
Date	scheduledate	The intended date and time of sending the SMS. Null for immediate sending. This value is used for all of the SMSs to be sent in this request.
String[]	recipientnames	The names of the recipients. Optional. This field is useful if names of recipients are to be stored in your Sent Messages. If not used, a null should be passed for this parameter. Otherwise, the number of elements in this array should equal the number of elements in the recipientMSISDNwithCC array. The name in each element in the array corresponds to the phone number in the recipientMSISDNwithCC at the corresponding index.
String[]	recipientMSISDNwithCC	The phone numbers of the recipients (with country code). Each element in the array should be a valid recipient phone number. The number of elements of this array MUST match the number of elements in the messagetext array. The phone number should not include any leading '0's or '+' symbols.
String[]	messagetext	The text content of the SMS for each recipient, with each index corresponding to the phone number in the corresponding entry

	in recipientMSISDNwithCC at the same index. The number of elements of this array MUST match the number of elements in the messagetext array.
--	----------------------------------------------------------------------------------------------------------------------------------------------

It returns an object with the following parameters.

Parameter Name	Description								
CallSuccess	If the common checks and authentication for the request are valid.								
CallErrorMessage	Error message if any occurred while authenticating the request.								
SomeMessagesPosted	True if the call was a success, and at least one of the recipients was validly posted to the queue. Should not be referred to if CallSuccess is false.								
MessagesPostedCount	Number of messages that were successfully posted to the queue. Will be 0 if CallSuccess is false.								
MultiSMSValidList	An array containing the details of the recipients to whom the message was successfully posted. Will be null if CallSuccess is false.								
	<table border="1"> <thead> <tr> <th>Parameter Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>RecipientName</td> <td>Name of the recipient</td> </tr> <tr> <td>RecipientPhoneNo</td> <td>MSISDN of the recipient</td> </tr> <tr> <td>OutMessageID</td> <td>Server generated unique ID representing an SMS successfully posted to the server.</td> </tr> </tbody> </table>	Parameter Name	Description	RecipientName	Name of the recipient	RecipientPhoneNo	MSISDN of the recipient	OutMessageID	Server generated unique ID representing an SMS successfully posted to the server.
Parameter Name	Description								
RecipientName	Name of the recipient								
RecipientPhoneNo	MSISDN of the recipient								
OutMessageID	Server generated unique ID representing an SMS successfully posted to the server.								
MulltiSMSInvalidList	An array containing the details of the recipients to whom the message could not be posted. Will be null if CallSuccess is false.								
	<table border="1"> <thead> <tr> <th>Parameter Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>RecipientName</td> <td>Name of the recipient</td> </tr> <tr> <td>RecipientPhoneNo</td> <td>MSISDN of the recipient</td> </tr> <tr> <td>ErrorMessage</td> <td>Error message while posting the SMS for this recipient.</td> </tr> </tbody> </table>	Parameter Name	Description	RecipientName	Name of the recipient	RecipientPhoneNo	MSISDN of the recipient	ErrorMessage	Error message while posting the SMS for this recipient.
Parameter Name	Description								
RecipientName	Name of the recipient								
RecipientPhoneNo	MSISDN of the recipient								
ErrorMessage	Error message while posting the SMS for this recipient.								

sendMultiSMS_String

This function is similar to the one above. The parameters passed to the function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.
String	batchname	Any unique name to identify the batch of SMS.
String	sendername	The name or shortcode that should appear as

		the sender of the SMS. This name or shortcode should be approved for your account. This value is used for all of the SMSs to be sent in this request.
boolean	sendernameisalpha	Pass true if the sendername parameter is an alphanumeric name; false if it is a shortcode. This value is used for all of the SMSs to be sent in this request.
boolean	requestreceipt	If the user wants a delivery report. This value is used for all of the SMSs to be sent in this request.
boolean	isflash	If the SMS is intended to be a flash message. This value is used for all of the SMSs to be sent in this request.
Date	scheduledate	The intended date and time of sending the SMS. Null for immediate sending. This value is used for all of the SMSs to be sent in this request.
String[]	recipientnames	The names of the recipients. Optional. This field is useful if names of recipients are to be stored in your Sent Messages. If not used, a null should be passed for this parameter. Otherwise, the number of elements in this array should equal the number of elements in the recipientMSISDNwithCC array. The name in each element in the array corresponds to the phone number in the recipientMSISDNwithCC at the corresponding index.
String[]	recipientMSISDNwithCC	The phone numbers of the recipients (with country code). Each element in the array should be a valid recipient phone number. The number of elements of this array MUST match the number of elements in the messagetext array. The phone number should not include any leading '0's or '+' symbols.
String[]	messagetext	The text content of the SMS for each recipient, with each index corresponding to the phone number in the corresponding entry in recipientMSISDNwithCC at the same index. The number of elements of this array MUST match the number of elements in the messagetext array.

It returns a String that is generated from the parameters of an object. The parameters and the string are as follows.

Parameter Name	Description
CallSuccess	If the common checks and authentication for the request are valid.
CallErrorMessage	Error message if any occurred while authenticating the request.
SomeMessagesPosted	True if the call was a success, and at least one of the

	recipients was validly posted to the queue. Should not be referred to if CallSuccess is false.								
MessagesPostedCount	Number of messages that were successfully posted to the queue. Will be 0 if CallSuccess is false.								
MultiSMSValidList	An array containing the details of the recipients to whom the message was successfully posted. Will be null if CallSuccess is false.								
	<table border="1"> <thead> <tr> <th>Parameter Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>RecipientName</td> <td>Name of the recipient</td> </tr> <tr> <td>RecipientPhoneNo</td> <td>MSISDN of the recipient</td> </tr> <tr> <td>OutMessageID</td> <td>Server generated unique ID representing an SMS successfully posted to the server.</td> </tr> </tbody> </table>	Parameter Name	Description	RecipientName	Name of the recipient	RecipientPhoneNo	MSISDN of the recipient	OutMessageID	Server generated unique ID representing an SMS successfully posted to the server.
Parameter Name	Description								
RecipientName	Name of the recipient								
RecipientPhoneNo	MSISDN of the recipient								
OutMessageID	Server generated unique ID representing an SMS successfully posted to the server.								
MulltiSMSInvalidList	An array containing the details of the recipients to whom the message could not be posted. Will be null if CallSuccess is false.								
	<table border="1"> <thead> <tr> <th>Parameter Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>RecipientName</td> <td>Name of the recipient</td> </tr> <tr> <td>RecipientPhoneNo</td> <td>MSISDN of the recipient</td> </tr> <tr> <td>ErrorMessage</td> <td>Error message while posting the SMS.</td> </tr> </tbody> </table>	Parameter Name	Description	RecipientName	Name of the recipient	RecipientPhoneNo	MSISDN of the recipient	ErrorMessage	Error message while posting the SMS.
Parameter Name	Description								
RecipientName	Name of the recipient								
RecipientPhoneNo	MSISDN of the recipient								
ErrorMessage	Error message while posting the SMS.								

Eg sendMultiSMS_String Result: "T","Success","2","2","T","Success","6458960","1","F","You do not have sufficient credit to send messages.,"6458962","1"

getAccountInfo

This function retrieves information of your account on the server. The parameters passed to the function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.

It returns an object with the following parameters.

Parameter Name	Description
CallSuccess	True: The call to the web service is successful. The amounts fields will contain information about your account. False: The call to the web service has failed, probably due to authentication issue or due to invalid username or password. Only the ErrMsg field should be used; the values in the other fields must be ignored.
ErrorMessage	The string value of the Error Message to be passed if the callSuccess is False, i.e. the user authentication fails.
AmountPresent	Current credit amount of the organization
Currency	Currency of all the amounts returned in this response
NumberOfHistoryMessages	No. of messages sent in the past 30 days
AmountUsedByUsersForSMS	Amount used by logins of the account for sending sms messages.
AmountUsedBySubaccountsForSMS	Amount used by sub-accounts of the account for sending sms messages.
ProfitFromSubaccountsForSMS	Profits received from sub-accounts due to sub-accounts' use of sms messages.
AmountUsedByUsersForNumerator	Amount used by logins of the account for Numerator.
AmountUsedBySubaccountsForNumerator	Amount used by sub-accounts of the account for Numerator.
ProfitFromSubaccountsForNumerator	Profits received from sub-accounts due to sub-accounts for Numerator.

Notes:

Some amounts are values accumulated throughout the lifetime of the account, i.e. they are never reset.

For e.g. AmountUsedByUsersForSMS is the total amount of all credits ever used for sending sms.

getInboxMessageSingle

This function retrieves information about an incoming message.

The parameters passed to the function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.

It returns an object with the following parameters.

Parameter Name	Description
MessagePresent	TRUE: Indicates that a message is available for this function call FALSE: Indicates that no message is present for this function call
HandsetPhoneNo	The sender's telephone number from which the message was received.
ShortCode	The short-code to which the message was received.
MessageContent	The content of the message received.
MessageReceivedAt	The Date at which the message was received in the inbox.
NoOfParts	The number of parts the message was received in.
CallSuccess	True: if the call to the method was successful. False: if the call to the method returned errors probably due to authentication issues.
ErrorMessage	The error message returned when the call to the method has failed.

getInboxMessageMulti

This function retrieves information about multiple incoming messages.

The parameters passed to the function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.
int	number of messages	The number of messages about which the information is required. Note that the maximum value for this field is 100 and the minimum is 1.

It returns an object with the following parameters.

Parameter Name	Description												
MessageList	An array containing message objects. Will be null if CallSuccess is false.												
	Each message object contains the following parameters: <table border="1" data-bbox="472 1016 1468 1266"> <thead> <tr> <th>Parameter Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>HandsetPhoneNo</td> <td>Telephone no. of the sender.</td> </tr> <tr> <td>ShortCode</td> <td>Short-code to which the message was sent.</td> </tr> <tr> <td>MessageContent</td> <td>Content of the message received.</td> </tr> <tr> <td>MessageReceivedAt</td> <td>Date on which the message was received.</td> </tr> <tr> <td>NoOfParts</td> <td>No. of parts the message was received in.</td> </tr> </tbody> </table>	Parameter Name	Description	HandsetPhoneNo	Telephone no. of the sender.	ShortCode	Short-code to which the message was sent.	MessageContent	Content of the message received.	MessageReceivedAt	Date on which the message was received.	NoOfParts	No. of parts the message was received in.
Parameter Name	Description												
HandsetPhoneNo	Telephone no. of the sender.												
ShortCode	Short-code to which the message was sent.												
MessageContent	Content of the message received.												
MessageReceivedAt	Date on which the message was received.												
NoOfParts	No. of parts the message was received in.												
CallSuccess	True: if the call to the method was successful. False: if the call to the method returned errors. This could be due to any of the following reasons: (a) authentication issues (b) invalid number of messages specified												
ErrorMessage	The error message returned when the call to the method has failed.												

getReceiptSingle

This function retrieves information about a receipt of a sent message.

The parameters passed to the function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.

It returns an object with the following parameters.

Parameter Name	Description
ReceiptPresent	TRUE: Indicates that a receipt is available for this function call FALSE: Indicates that no receipt is present for this function call
ReceiptID	This field uniquely identifies the receipt
HandsetPhoneNo	The recipient's telephone number to which the message was sent.
ShortCode	The short-code that was used in sending the message to the recipient.
OutMessageID	This field is a unique 64-bit message id which correlates to this receipt id.
MessagePostedOn	The Date at which the message was posted to be sent to the recipient. As the posted date could also have been a date which was scheduled to a future date.
MessageReceivedOn	The Date at which the message was received by the recipient.
MessageDeliveryStatus	The delivery status of the receipt, these are of four types: (a)Delivered: Message was successfully delivered (b)Undelivered: Message was not delivered (c)Expired: Message expired (d)Unknown: Message status unknown
CallSuccess	True: if the call to the method was successful. False: if the call to the method returned errors, probably due to authentication issues.
ErrorMessage	The error message returned when the call to the method has failed.

getReceiptMulti

This function retrieves information about multiple receipts of the messages sent.

The parameters passed to the function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.
int	number of receipts	The number of receipts about which the information is required. Note that the maximum value for this field is 100 and the minimum is 1.

It returns an object with the following parameters.

Parameter Name	Description																
ReceiptList	An array containing receipt objects. Will be null if CallSuccess is false.																
	Each receipt object contains the following parameters: <table border="1"> <thead> <tr> <th>Parameter Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ReceiptID</td> <td>The field uniquely identifies the receipt</td> </tr> <tr> <td>HandsetPhoneNo</td> <td>The recipient's telephone number to which the message was sent.</td> </tr> <tr> <td>ShortCode</td> <td>The short-code that was used in sending the message to the recipient.</td> </tr> <tr> <td>OutMessageID</td> <td>This field is a unique 64-bit message id which co-relates to this receipt id.</td> </tr> <tr> <td>MessagePostedOn</td> <td>The Date at which the message was posted to be sent to the recipient. As the posted date can also be a date which was scheduled to a future date.</td> </tr> <tr> <td>MessageReceivedOn</td> <td>The Date at which the message was received by the recipient.</td> </tr> <tr> <td>MessageDeliveryStatus</td> <td>The delivery status of the receipt, these are of four types: (a)Delivered: Message delivery successful (b)Undelivered: Message was not delivered (c)Expired: Message expired (d)Unknown: Message status unknown</td> </tr> </tbody> </table>	Parameter Name	Description	ReceiptID	The field uniquely identifies the receipt	HandsetPhoneNo	The recipient's telephone number to which the message was sent.	ShortCode	The short-code that was used in sending the message to the recipient.	OutMessageID	This field is a unique 64-bit message id which co-relates to this receipt id.	MessagePostedOn	The Date at which the message was posted to be sent to the recipient. As the posted date can also be a date which was scheduled to a future date.	MessageReceivedOn	The Date at which the message was received by the recipient.	MessageDeliveryStatus	The delivery status of the receipt, these are of four types: (a)Delivered: Message delivery successful (b)Undelivered: Message was not delivered (c)Expired: Message expired (d)Unknown: Message status unknown
Parameter Name	Description																
ReceiptID	The field uniquely identifies the receipt																
HandsetPhoneNo	The recipient's telephone number to which the message was sent.																
ShortCode	The short-code that was used in sending the message to the recipient.																
OutMessageID	This field is a unique 64-bit message id which co-relates to this receipt id.																
MessagePostedOn	The Date at which the message was posted to be sent to the recipient. As the posted date can also be a date which was scheduled to a future date.																
MessageReceivedOn	The Date at which the message was received by the recipient.																
MessageDeliveryStatus	The delivery status of the receipt, these are of four types: (a)Delivered: Message delivery successful (b)Undelivered: Message was not delivered (c)Expired: Message expired (d)Unknown: Message status unknown																
CallSuccess	True: if the call to the method was successful. False: if the call to the method returned errors. This could be due to any of the following reasons: (a)authentication issues (b)invalid number of messages specified																
ErrorMessage	The error message returned when the call to the method has failed.																

Methods Available for testing

testConnection

This function is used for testing connectivity to the server.

Parameter Type	Parameter Name	Description
none		

Parameter Type	Parameter Name	Description
String		It returns a string which includes the current date & time. This may change in future implementations; calling applications should not expect any particular format.

testAuthentication

This function is used for testing the authentication of the user to the server.

The parameters passed to the function are:

Parameter Type	Parameter Name	Description
String	username	The login name used to access the account.
String	password	The password used to access the account.

Parameter Type	Parameter Name	Description
String		It returns a String which describes the success or failure of the authentication.