

directSMS



SOAP API

February 2005

www.directsms.com.au

Table of Contents

- OVERVIEW..... 4**
- SOAP API INTRODUCTION..... 5**
- INTRODUCTION 5**
 - WHAT IS SOAP? 5
 - WHY SOAP?..... 5
 - WHAT IS WSDL? 5
 - WSDL DESCRIBES WEB SERVICES 6
 - THE SOAP FAULT ELEMENT..... 6
 - SOAP Fault Codes..... 6
- MSGGATEWAY SOAP SERVICE..... 7**
 - CONNECT () 8
 - SOAP Request 8
 - SOAP Response..... 8
 - SOAP Fault 9
 - CONNECTWITHLICENCE () 10
 - SOAP Request 10
 - SOAP Response..... 10
 - SOAP Fault 11
 - SENDERBRANDEDSMSMESSAGE () 12
 - SOAP Request 12
 - SOAP Response..... 13
 - SOAP Fault 13
 - SCHEDULEBRANDEDSMSMESSAGE () 15
 - SOAP Request 15
 - SOAP Response..... 16
 - SOAP Fault 16
 - SENDERBRANDEDSMSMESSAGES () 18
 - SOAP Request 18
 - SOAP Response..... 19
 - SOAP Fault 20
 - SENDTOWAYSMSMESSAGE () 21
 - SOAP Request 21
 - SOAP Response..... 22
 - SOAP Fault 22
 - SCHEDULETOWAYSMSMESSAGE () 24
 - SOAP Request 24
 - SOAP Response..... 25
 - SOAP Fault 25
 - SENDTOWAYSMSMESSAGES () 27
 - SOAP Request 27
 - SOAP Response..... 28
 - SOAP Fault 29
 - GETALLSMSMESSAGEREPLIES () 30
 - SOAP Request 30
 - SOAP Response..... 30
 - SOAP Fault 32
 - GETSMSMESSAGEREPLIES () 33
 - SOAP Request 33
 - SOAP Response..... 33
 - SOAP Fault 35
 - GETALLINBOUNDSMSMESSAGES () 36

directSMS SOAP API

SOAP Request 36
SOAP Response..... 36
SOAP Fault..... 37
GETINBOUNDSmsMESSAGES () 39
 SOAP Request 39
 SOAP Response..... 39
 SOAP Fault..... 41
GETCURRENTBALANCE () 42
 SOAP Request 42
 SOAP Response..... 42
 SOAP Fault 43
DISCONNECT () 44
 SOAP Request 44
 SOAP Response..... 44
 SOAP Fault 45

OVERVIEW

This document is intended for developers who wish to develop SOAP enabled applications that make use of directSMS' SMS Gateway to send and receive SMS messages.

This document will concentrate on the **SmsGateway** SOAP service and will detail the various operations available through this service. The operations of the service act as single units of work for transaction purposes.

In the event of any errors occurring during any operation, a SOAP fault object is returned detailing the error event.

SOAP API INTRODUCTION

Introduction

What is SOAP?

- SOAP stands for **Simple Object Access Protocol**
 - SOAP is a **communication protocol**
 - SOAP is for communication **between applications**
 - SOAP is a format for **sending messages**
 - SOAP is designed to communicate **via the Internet**
 - SOAP is **platform independent**
 - SOAP is **language independent**
 - SOAP is **based on XML**
 - SOAP is **simple and extensible**
 - SOAP allows you to **get around firewalls**
 - SOAP will be developed as a **W3C standard**
-

Why SOAP?

It is important for application development to allow Internet communication between programs. Today's applications communicate using Remote Procedure Calls (RPC) between objects like DCOM and CORBA, but HTTP was not designed for this. RPC represents a compatibility and security problem; firewalls and proxy servers will normally block this kind of traffic.

A better way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this.

SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

What is WSDL?

- WSDL stands for Web Services Description Language
- WSDL is written in XML
- WSDL is an XML document
- WSDL is used to describe Web services
- WSDL is also used to locate Web services

- WSDL is not yet a W3C standard

WSDL Describes Web Services

WSDL stands for Web Services Description Language.

WSDL is a document written in XML. The document describes a Web service. It specifies the location of the service and the operations (or methods) the service exposes.

The SOAP Fault Element

An error message from a SOAP message is carried inside a Fault element.

If a Fault element is present, it must appear as a child element of the Body element. A Fault element can only appear once in a SOAP message.

The SOAP Fault element has the following sub elements:

| Sub Element | Description |
|--------------------|--|
| <faultcode> | A code for identifying the fault |
| <faultstring> | A human readable explanation of the fault |
| <faultactor> | Information about who caused the fault to happen |
| <detail> | Holds application specific error information related to the Body element |

SOAP Fault Codes

The faultcode values defined below must be used in the faultcode element when describing faults:

| Error | Description |
|-----------------|--|
| VersionMismatch | Found an invalid namespace for the SOAP Envelope element |
| MustUnderstand | An immediate child element of the Header element, with the mustUnderstand attribute set to "1", was not understood |
| Client | The message was incorrectly formed or contained incorrect information |
| Server | There was a problem with the server so the message could not proceed |

SmsGateway SOAP Service

SmsGateway is a SOAP web service published by directSMS in order to enable our customers' systems to send and receive SMS messages.

The service is available on the following URLs both over HTTP and HTTPS (HTTP over SSL using public key encryption).

HTTP

Service URL: <http://api.directsms.com.au/s3/soap/SmsGateway>

HTTPS

Service URL: <https://api.directsms.com.au/s3/soap/SmsGateway>

The WSDL document defining the various operations available through the **SmsGateway** service can be found at:

<http://www.directsms.com.au/info/api/SmsGateway-HTTP.wsdl>

Similarly, the WSDL utilizing the more secure HTTPS protocol can be found at:

<http://www.directsms.com.au/info/api/SmsGateway-HTTPS.wsdl>

This WSDL can be used by any SOAP toolkit to generate the client side stubs used in accessing the service. The deployment over SSL provides a more secure interface for clients.

The following is a list of all the operations available through this web service.

connect ()

The **connect** request contains a regular directSMS username and password. If the credentials passed in the request match an active directSMS user account, a connection id is returned. This connection id is used during all subsequent operations. If the credentials are invalid or the SMS gateway is experiencing any difficulties, an Exception (translated to a SOAP Fault object is returned instead).

*All other operations will fail until you have successfully **connected** to the gateway.* The connection Id returned has to be used in all future operations with the SMS gateway.

If the user account used is not set up for API access, an error will be returned.

In order to use the API, you can either contact the directSMS Support team to enable this feature on your account. This will activate your 60 day API trial.

SOAP Request

The request is composed of a directSMS username and password.

```

/*
 * Structure to encapsulate the request to open a new SMS gateway
 * connection
 */
ConnectRequest
{
    String: username // directSMS username
    String: password // directSMS password
}

```

A sample request SOAP message is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:connect>
      <username xsi:type="xsd:string">j_smith</username>
      <password xsi:type="xsd:string">Pa55w0rD</password>
    </ns4:connect>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message for this operation is composed of a directSMS connection id, which is a 32 character unique identifier that identifies the given user's Gateway connection and is used in every subsequent operation with the Gateway.

```

/*
 * Data structure to encapsulate a successful response to a
 * connection request
 */
ConnectResponse

```



```
{
    String: connectionId    // Connection identifier for an open
                          // SMS engine connection
}
```

A sample response message is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:connectResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <connectReturn xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectReturn>
    </ns1:connectResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the username and password passed in the connect request were invalid.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException: invalid login
        credentials
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

connectWithLicence ()

The **connectWithLicence** request contains a regular directSMS username and password, as well as a valid licence key. If the credentials passed in the request match an active directSMS user account, and a valid enterprise licence, a connection id is returned. This connection id is used during all subsequent operations. If the credentials are invalid or the gateway is experiencing any difficulties, an Exception (translated to a SOAP Fault object) is returned instead).

*All other operations will fail until you have successfully **connected** to the gateway. The connection Id returned has to be used in all future operations with the gateway.*

If the user account used is not set up for API access, an error will be returned.

In order to use the API, you can either contact the directSMS Support team to enable this feature on your account.

SOAP Request

The request is composed of a directSMS username and password.

```

/*
 * Structure to encapsulate the request to open a new Gateway
 * connection
 */
ConnectWithLicenceRequest
{
    String: username           // directSMS username
    String: password          // directSMS password
    String: licenceKey        // directSMS licence key
}

```

A sample request SOAP message is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:connectWithLicence>
      <username xsi:type="xsd:string">j_smith</username>
      <password xsi:type="xsd:string">Pa55w0rD</password>
      <licenceKey xsi:type="xsd:string">
        0d058ff1-3d3d1c26-a337ef08-cbea45f5
      </licenceKey>
    </ns4:connectWithLicence>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message for this operation is composed of a directSMS connection id, which is a 32 character unique identifier that identifies the given user's gateway connection and is used in every subsequent operation with the SMS gateway.

```

/*
 * Data structure to encapsulate a successful response to a
 * connection request
 */
ConnectWithLicenceResponse
{
    String: connectionId    // Connection identifier for an open
                          // gateway connection
}

```

A sample response message is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:connectWithLicenceResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <connectWithLicenceReturn xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectWithLicenceReturn>
    </ns1:connectWithLicenceResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the username and password passed in the connect request were invalid.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException: invalid licence key
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

sendBrandedSmsMessage ()

The **sendBrandedSmsMessage** operation is used to send a branded 1-way SMS to one or more mobile phones. To use this method you must first call the **connect()** operation to authenticate your login credentials against the server and to obtain a connection id. If the send operation is successful, a unique 32 character message identifier is returned to enable the tracking of the sent messages in directSMS' systems.

SOAP Request

The sendBrandedSmsMessageRequest comprising the XML request message for the send operation is composed of a connection id obtained in an earlier call to the connect operation and a BrandedSmsMessage structure. This in turn is made up of a collection of Strings and String arrays. This is done to keep messaging structures simple for the sake of interoperability.

Mobile phones can be passed in any valid format, e.g. 0412 345 678, 61 412 345 678 or +61 412 345 678.

```

/*
 * Structure to model a Branded 1-way SMS message
 */
BrandedSmsMessage
{
    String:    message    // SMS message text
    String:    senderId  // Alphanumeric sender id. The sender id
                        // must be less than 12 characters and
                        // should not contain characters outside
                        // of a-z, A-Z, . or 0-9. The sender id
                        // can be set to a mobile number. This
                        // would allow recipients to reply back
                        // the number set. The replies however can
                        // not be tracked by directSMS' gateway
    String[]:  mobiles   // Array of valid mobile phone numbers to
                        // send this message to
}

/*
 * Request to send a branded 1-way SMS message
 */
SendBrandedSmsMessageRequest
{
    String:    connectionId // Connection id for an open
                        // SMS gateway connection
    BrandedSmsMessage: brandedSms // SMS message to send
}

```

A sample SOAP request is shown bellow to send the branded message “this is a test message” with the Sender Id of “directSMS” to 3 mobile phones.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:sendBrandedSmsMessage>

```

```

<connectionId xsi:type="xsd:string">
  f4b9ec30ad9f68f89b29639786cb62ef
</connectionId>
<brandedSms xsi:type="ns4:BrandedSmsMessage">
  <message xsi:type="xsd:string">
    this is a test message
  </message>
  <senderId xsi:type="xsd:string">directSMS</senderId>
  <mobiles xsi:type="SOAP-ENC:Array"
    SOAP-ENC:arrayType="xsd:string[3]"
    SOAP-ENC:offset="[0]">
    <item xsi:type="xsd:string">0407 263 977</item>
    <item xsi:type="xsd:string">0414 574 496</item>
    <item xsi:type="xsd:string">0411 206 954</item>
  </mobiles>
</brandedSms>
</ns4:sendBrandedSmsMessage>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message will contain another 32 character String that uniquely identifies the 1-way SMS just submitted in order to track messages in directSMS' systems.

```

/*
 * Successful response to a sendBrandedSmsMessage operation
 */
SendBrandedSmsMessageResponse
{
  String: smsId // directSMS id for the submitted SMS message
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:sendBrandedSmsMessageResponse
      soapenv:encodingStyle=" http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <sendBrandedSmsMessageReturn xsi:type="xsd:string">
        0e65972dce68dad4d52d063967f0a705
      </sendBrandedSmsMessageReturn>
    </ns1:sendBrandedSmsMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned contains a faultstring element which should describe exactly what went wrong. In the example below, the user account used does not have sufficient SMS credits to send the submitted message.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>

```

directSMS SOAP API

```
<faultstring>  
  com.directsms.s3.service.sms.SmsServiceException:  
  insufficient credit to send message(s)  
</faultstring>  
<detail/>  
</soapenv:Fault>  
  
</soapenv:Body>  
</soapenv:Envelope>
```

scheduleBrandedSmsMessage ()

The **scheduleBrandedSmsMessage** operation is used to schedule the sending of a 1-way SMS to one or more mobile phones at a later time. To use this method you must first call the **connect()** operation to authenticate your login credentials against the server and to obtain a connection id. If the schedule operation is successful, a unique 32 character message identifier is returned to enable the tracking of the sent messages in directSMS' systems.

SOAP Request

The scheduleBrandedSmsMessageRequest comprising the XML request message for the schedule operation is composed of a connection id obtained in an earlier call to the connect operation, a BrandedSmsMessage structure and the date/time to send the message. The date/time is passed as a long integer representing the date/time as the number of seconds after the Unix Epoch (January 1 1970 00:00:00 GMT). This ensures time zone independence.

All structures are made up of a collection of Strings and String arrays. This is done to keep messaging structures simple for the sake of interoperability.

Mobile phones can be passed in any valid format, e.g. 0412 345 678, 61 412 345 678 or +61 412 345 678.

```

/*
 * Structure to model a Branded 1-way SMS message
 */
BrandedSmsMessage
{
    String:    message    // SMS message text
    String:    senderId   // Alphanumeric sender id. The sender id
                        // must be less than 12 characters and
                        // should not contain characters outside
                        // of a-z, A-Z, . or 0-9. The sender id
                        // can be set to a mobile number. This
                        // would allow recipients to reply back
                        // the number set. The replies however can
                        // not be tracked by directSMS' gateway
    String[]:  mobiles    // Array of valid mobile phone numbers to
                        // send this message to
}

/*
 * Request to send a branded 1-way SMS message
 */
ScheduleBrandedSmsMessageRequest
{
    String:    connectionId // Connection id for an open
                        // SMS gateway connection
    BrandedSmsMessage: brandedSms // SMS message to send
    long:      timestamp     // Date/Time to send the SMS
                        // message as the number of
                        // seconds after the Unix Epoch
}

```

A sample SOAP request is shown below to send the branded message "this is a test message" with the Sender Id of "directSMS" to 3 mobile phones.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:scheduleBrandedSmsMessage>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
      <brandedSms xsi:type="ns4:BrandedSmsMessage">
        <message xsi:type="xsd:string">
          This is a test message to go out at 11AM, January 1, 2006
          (Eastern Standard Time)
        </message>
        <senderId xsi:type="xsd:string">directSMS</senderId>
        <mobiles xsi:type="SOAP-ENC:Array"
          SOAP-ENC:arrayType="xsd:string[3]"
          SOAP-ENC:offset="[0]">
          <item xsi:type="xsd:string">0407 263 977</item>
          <item xsi:type="xsd:string">0414 574 496</item>
          <item xsi:type="xsd:string">0411 206 954</item>
        </mobiles>
      </brandedSms>
      <timestamp xsi:type="xsd:long">
        1164931200
      </timestamp>
    </ns4:scheduleBrandedSmsMessage>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Response

The response message will contain another 32 character String that uniquely identifies the branded 1-way SMS just submitted in order to track messages in directSMS' systems.

```
/*
 * Successful response to a scheduleBrandedSmsMessage operation
 */
ScheduleBrandedSmsMessageResponse
{
    String: smsId // directSMS id for the submitted branded 1-way SMS
                // message
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:scheduleBrandedSmsMessageResponse
      soapenv:encodingStyle=" http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <scheduleBrandedSmsMessageReturn xsi:type="xsd:string">
        0e65972dce68dad4d52d063967f0a705
      </scheduleBrandedSmsMessageReturn>
    </ns1:scheduleBrandedSmsMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned contains a faultstring element which should describe exactly what went wrong. In

directSMS SOAP API

the example below, the user account used does not have sufficient SMS credits to send the submitted message.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        insufficient credit to send message(s)
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

sendBrandedSmsMessages ()

The **sendBrandedSmsMessages** operation is used to send a set of branded 1-way SMS messages; each message going out to one or more mobile phones. As with all other operations you must first call the **connect()** operation to authenticate your login credentials against the server and to obtain a connection id.

If the send operation is successful, an array of unique 32 character message identifiers is returned to enable the tracking of the sent messages in directSMS' systems.

SOAP Request

The sendBrandedSmsMessagesRequest comprising the XML request message for the send operation is composed of a the connection id obtained in an earlier call to the connect operation and an array of BrandedSmsMessage structures. These in turn are made up of a collection of Strings and String arrays. This is done to keep messaging structures simple for the sake of interoperability.

Mobile phones can be passed in any valid format, e.g. 0412 345 678, 61 412 345 678 or +61 412 345 678.

```

/*
 * Structure to model a Branded 1-way SMS message
 */
BrandedSmsMessage
{
    String:    message    // SMS message text
    String:    senderId   // Alphanumeric sender id. The sender id
                        // must be less than 12 characters and
                        // should not contain characters outside
                        // of a-z, A-Z, . or 0-9. The sender id
                        // can be set to a mobile number. This
                        // would allow recipients to reply back
                        // the number set. The replies however can
                        // not be tracked by directSMS' gateway
    String[]:  mobiles    // Array of valid mobile phone numbers to
                        // send this message to
}

/*
 * Request to send a set of branded 1-way SMS messages
 */
SendBrandedSmsMessagesRequest
{
    String:    connectionId // Connection id for
                        // an open SMS
                        // gateway connection
    BrandedSmsMessage[]:    brandedSms // SMS message to
                        // send
}

```

A sample SOAP request is shown bellow to send the branded message “this is a test message” with the Sender Id of “directSMS” to 3 mobile phones.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns4="urn:SmsGateway">
<SOAP-ENV:Body>

  <ns4:sendBrandedSmsMessages>
    <connectionId xsi:type="xsd:string">
      42a0e188f5033bc65bf8d78622277c4e
    </connectionId >
    <messages xsi:type="SOAP-ENC:Array"
      SOAP-ENC:arrayType="ns4:BrandedSmsMessage[2]" SOAP-ENC:offset=" [0]">
      <brandedSms xsi:type="ns4:BrandedSmsMessage">
        <message xsi:type="xsd:string">
          this is a test message at 2004-06-08 20:34:00
        </message>
        <senderId xsi:type="xsd:string">directsms</senderId>
        <mobiles xsi:type="SOAP-ENC:Array"
          SOAP-ENC:arrayType="xsd:string[3]" SOAP-ENC:offset=" [0]">
          <item xsi:type="xsd:string">0407 263 977</item>
          <item xsi:type="xsd:string">0414 574 496</item>
          <item xsi:type="xsd:string">0411 206 954</item>
        </mobiles>
      </brandedSms>
      <brandedSms xsi:type="ns4:BrandedSmsMessage">
        <message xsi:type="xsd:string">
          this is a test message at 2004-06-08 20:34:00
        </message>
        <senderId xsi:type="xsd:string">directsms</senderId>
        <mobiles xsi:type="SOAP-ENC:Array"
          SOAP-ENC:arrayType="xsd:string[3]" SOAP-ENC:offset=" [0]">
          <item xsi:type="xsd:string">0407 263 977</item>
          <item xsi:type="xsd:string">0414 574 496</item>
          <item xsi:type="xsd:string">0411 206 954</item>
        </mobiles>
      </brandedSms>
    </messages>
  </ns4:sendBrandedSmsMessages>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message will contain an array of 32 character Strings that uniquely identify the branded 1-way SMS messages submitted in order to track the messages in directSMS' systems.

```

/*
 * Successful response to a sendBrandedSmsMessages operation
 */
SendBrandedSmsMessagesResponse
{
    String[]: smsIds    // directSMS ids for the submitted branded
                       // SMS messages
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>

    <ns1:sendBrandedSmsMessagesResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <sendBrandedSmsMessagesReturn xsi:type="soapenc:Array"
        soapenc:arrayType="xsd:string[2]"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <item>142949df56ea8ae0be8b5306971900a4</item>
        <item>d34ab169b70c9dcd35e62896010cd9ff</item>
      </sendBrandedSmsMessagesReturn>
    </ns1:sendBrandedSmsMessagesResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

```
</soapenv:Body>  
</soapenv:Envelope>
```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned contains a faultstring element which should describe exactly what went wrong. In the example below, the user account used does not have sufficient SMS credits to send the submitted message.

```
<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  <soapenv:Body>  
    <soapenv:Fault>  
      <faultcode>  
        soapenv:Server.userException  
      </faultcode>  
      <faultstring>  
        com.directsms.s3.service.sms.SmsServiceException:  
        insufficient credit to send message(s)  
      </faultstring>  
      <detail/>  
    </soapenv:Fault>  
  </soapenv:Body>  
</soapenv:Envelope>
```

sendTwoWaySmsMessage ()

This operation is used to send a 2-way SMS to one or more mobile phones. Once sent, recipients of the message can reply and the directSMS gateway will route response messages back to your directSMS account to allow you to view them.

Replies can be retrieved through the various APIs offered by directSMS, or directly through the website at <http://www.directsms.com.au>

SOAP Request

The request SOAP message is composed of the connection id obtained from an earlier connect operation and a TwoWaySmsMessage structure. Which in turn is made of a collection of Strings and String arrays.

The TwoWaySmsMessage allows for clients to include a messageId which is an identifier from the client's system to help them link SMS replies to the original messages submitted through this operation. Again, clients must successfully connect to the gateway before they are allowed to use this operation.

```

/*
 * Data structure to model a 2-way SMS message
 */
TwoWaySmsMessage
{
    String:    message    // SMS message text, must be 160 characters
                  // or less
    String:    messageId // An optional parameter that can be used to
                  // identify this message in the client's
                  // system/application
    String[]:  mobiles    // Array of mobile numbers to send this
                  // message to
}

/*
 * Request to send a 2-way SMS message
 */
SendTwoWaySmsMessageRequest
{
    String:          connectionId // Connection id for an open
                  // SMS gateway connection
    TwoWaySmsMessage: twoWaySms    // SMS message to send
}

```

The example SOAP message shown below contains a value for the optional messageId component which can be used in conjunction with the **getSmsReplies** operation..

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:sendTwoWaySmsMessage>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
    </ns4:sendTwoWaySmsMessage>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

<twoWaySms xsi:type="ns4:TwoWaySmsMessage">
  <message xsi:type="xsd:string">
    this is a 2-way test message with a message id
  </message>
  <messageId>a message id</messageId>
  <mobiles xsi:type="SOAP-ENC:Array"
    SOAP-ENC:arrayType="xsd:string[3]"
    SOAP-ENC:offset="[0]">
    <item xsi:type="xsd:string">0407 263 977</item>
    <item xsi:type="xsd:string">0414 574 496</item>
    <item xsi:type="xsd:string">0411 206 954</item>
  </mobiles>
</twoWaySms>
</ns4:sendTwoWaySmsMessage>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message will contain a 32 character String that uniquely identifies the 2-way SMS submitted in order to track messages in directSMS' systems. And also to provide the client with a confirmation that the message was submitted successfully.

```

/*
 * Successful response to a sendTwoWaySmsMessage operation
 */
SendTwoWaySmsMessageResponse
{
    String: smsId // directSMS id for the submitted 2-way message
}

```

The sample SOAP message below shows a successful 2-way SMS submitted.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:sendTwoWaySmsMessageResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <sendTwoWaySmsMessageReturn xsi:type="xsd:string">
        33e75ff09dd601bbe69f351039152189
      </sendTwoWaySmsMessageReturn>
    </ns1:sendTwoWaySmsMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the user account used does not have sufficient SMS credits to send the submitted message.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

directSMS SOAP API

```
</faultcode>
<faultstring>
  com.directsms.s3.service.sms.SmsServiceException:
  insufficient credit to send message(s)
</faultstring>
<detail/>
</soapenv:Fault>

</soapenv:Body>
</soapenv:Envelope>
```

scheduleTwoWaySmsMessage ()

The **scheduleTwoWaySmsMessage** operation is used to schedule the sending of a 2-way SMS to one or more mobile phones at a later time. To use this method you must first call the **connect()** operation to authenticate your login credentials against the server and to obtain a connection id. If the schedule operation is successful, a unique 32 character message identifier is returned to enable the tracking of the sent messages in directSMS' systems.

SOAP Request

The scheduleTwoWaySmsMessageRequest comprising the XML request message for the schedule operation is composed of a connection id obtained in an earlier call to the connect operation, a TwoWaySmsMessage structure and the date/time to send the message. The date/time is passed as a long integer representing the date/time as the number of seconds after the Unix Epoch (January 1 1970 00:00:00 GMT). This ensures time zone independence.

All structures are made up of a collection of Strings and String arrays. This is done to keep messaging structures simple for the sake of interoperability.

Mobile phones can be passed in any valid format, e.g. 0412 345 678, 61 412 345 678 or +61 412 345 678.

```

/*
 * Data structure to model a 2-way SMS message
 */
TwoWaySmsMessage
{
    String:    message    // SMS message text, must be 160 characters
                  // or less
    String:    messageId // An optional parameter that can be used to
                  // identify this message in the client's
                  // system/application
    String[]:  mobiles    // Array of mobile numbers to send this
                  // message to
}

/*
 * Request to send a branded 1-way SMS message
 */
ScheduleBrandedSmsMessageRequest
{
    String:          connectionId // Connection id for an open
                  // SMS gateway connection
    TwoWaySmsMessage: twoWaySms    // SMS message to send
    long:            timestamp      // Date/Time to send the SMS
                  // message as the number of
                  // seconds after the Unix Epoch
}

```

A sample SOAP request is shown bellow to send the branded message "this is a test message" with the Sender Id of "directSMS" to 3 mobile phones.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"

```



```

xmlns:ns4="urn:SmsGateway"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
<SOAP-ENV:Body>

  <ns4:scheduleTwoWaySmsMessage>
    <connectionId xsi:type="xsd:string">
      f4b9ec30ad9f68f89b29639786cb62ef
    </connectionId>
    <twoWaySms xsi:type="ns4:TwoWaySmsMessage">
      <message xsi:type="xsd:string">
        This is a 2-way test message with a message id, scheduled
        To go out at 11AM, January 1, 2006 (Eastern Standard Time)
      </message>
      <messageId>a message id</messageId>
      <mobiles xsi:type="SOAP-ENC:Array"
        SOAP-ENC:arrayType="xsd:string[3]"
        SOAP-ENC:offset="[0]">
        <item xsi:type="xsd:string">0407 263 977</item>
        <item xsi:type="xsd:string">0414 574 496</item>
        <item xsi:type="xsd:string">0411 206 954</item>
      </mobiles>
    </twoWaySms>
    <timestamp xsi:type="xsd:long">
      1164931200
    </timestamp>
  </ns4:scheduleTwoWaySmsMessage>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message will contain another 32 character String that uniquely identifies the branded 1-way SMS just submitted in order to track messages in directSMS' systems.

```

/*
 * Successful response to a scheduleTwoWaySmsMessage operation
 */
ScheduleTwoWaySmsMessageResponse
{
    String: smsId // directSMS id for the submitted SMS message
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>

    <ns1:scheduleTwoWaySmsMessageResponse
      soapenv:encodingStyle=" http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <sendTwoWaySmsMessageReturn xsi:type="xsd:string">
        0e65972dce68dad4d52d063967f0a705
      </sendTwoWaySmsMessageReturn>
    </ns1:scheduleTwoWaySmsMessageResponse>

  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned contains a faultstring element which should describe exactly what went wrong. In the example below, the user account used does not have sufficient SMS credits to send the submitted message.

```

<?xml version="1.0" encoding="UTF-8"?>

```

directSMS SOAP API

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        insufficient credit to send message(s)
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

sendTwoWaySmsMessages ()

This operation is used to send a set of 2-way SMS messages, with each message going to one or more mobile phones. Once sent, recipients of the messages can reply and the directSMS gateway will route response messages back to your directSMS account to allow you to view them. Replies can be retrieved through the various APIs offered by directSMS, or directly through the website at <http://www.directsms.com.au>

SOAP Request

The request SOAP message is composed of the connection id obtained from an earlier connect operation and an array of TwoWaySmsMessage structures. These structures in turn are made of a collection of Strings and String arrays.

The TwoWaySmsMessage allows for clients to include a messageId which is an identifier from the client's system to help them link SMS replies to the original messages submitted through this operation. Again, clients must successfully connect to the gateway before they are allowed to use this operation.

```

/*
 * Data structure to model a 2-way SMS message
 */
TwoWaySmsMessage
{
    String:    message    // SMS message text, must be 160 characters
                  // or less
    String:    messageId // An optional parameter that can be used to
                  // identify this message in the client's
                  // system/application
    String[]:  mobiles    // Array of mobile numbers to send this
                  // message to
}

/*
 * Request to send a set of 2-way SMS message
 */
SendTwoWaySmsMessagesRequest
{
    String:    connectionId    // Connection id for an
                  // open SMS gateway
                  // connection
    TwoWaySmsMessage[]: twoWaySms    // SMS messages to send
}

```

The example SOAP request message shown below does not contain a value for the optional messageId component which can be used in conjunction with the **getSmsReplies** operation..

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway">
  <SOAP-ENV:Body>
    <ns4:sendTwoWaySmsMessages>
      <in0 xsi:type="xsd:string">42a0e188f5033bc65bf8d78622277c4e</in0>
      <in1 xsi:type="SOAP-ENC:Array"
        SOAP-ENC:arrayType="ns4:TwoWaySmsMessage [2]" SOAP-ENC:offset="[0]">

```

```

<twoWaySms xsi:type="ns4:TwoWaySmsMessage">
  <message xsi:type="xsd:string">
    this is a 2-way test message at 2004-06-08 20:34:00
  </message>
  <messageId/>
  <mobiles xsi:type="SOAP-ENC:Array"
    SOAP-ENC:arrayType="xsd:string[3]" SOAP-ENC:offset="[0]">
    <item xsi:type="xsd:string">0407 263 977</item>
    <item xsi:type="xsd:string">0414 574 496</item>
    <item xsi:type="xsd:string">0411 206 954</item>
  </mobiles>
</twoWaySms>
<twoWaySms xsi:type="ns4:TwoWaySmsMessage">
  <message xsi:type="xsd:string">
    this is a 2-way test message at 2004-06-08 20:34:00
  </message>
  <messageId/>
  <mobiles xsi:type="SOAP-ENC:Array"
    SOAP-ENC:arrayType="xsd:string[3]" SOAP-ENC:offset="[0]">
    <item xsi:type="xsd:string">0407 263 977</item>
    <item xsi:type="xsd:string">0414 574 496</item>
    <item xsi:type="xsd:string">0411 206 954</item>
  </mobiles>
</twoWaySms>
</in1>
</ns4:sendTwoWaySmsMessages>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message will contain a 32 character String that uniquely identifies the 2-way SMS submitted in order to track messages in directSMS' systems. And also to provide the client with a confirmation that the message was submitted successfully.

```

/*
 * Successful response to a sendTwoWaySmsMessage operation
 */
SendTwoWaySmsMessageResponse
{
    String: smsId // directSMS id for the submitted 2-way message
}

```

The sample SOAP message below shows both messages submitted were successfully accepted.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:sendTwoWaySmsMessagesResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <sendTwoWaySmsMessagesReturn
        xsi:type="soapenc:Array"
        soapenc:arrayType="xsd:string[2]"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <item>f7e6c85504ce6e82442c770f7c8606f0</item>
        <item>bf8229696f7a3bb4700cfddef19fa23f</item>
      </sendTwoWaySmsMessagesReturn>
    </ns1:sendTwoWaySmsMessagesResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the user account used does not have sufficient SMS credits to send the submitted message.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        insufficient credit to send message(s)
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

getAllSmsMessageReplies ()

The operation to fetch all "unread" SMS replies to ALL 2-way messages submitted in the past.

SOAP Request

The request contains a boolean flag to indicate to the directSMS gateway whether or not you want to mark the reply messages returned as "read". This stops them from being retrieved in future calls to this operation as well as **getSmsMessageReplies**.

```

/*
 * Request to retrieve all responses to 2-way SMS messages sent
 */
GetAllSmsMessageRepliesRequest
{
    String:    connectionId    // Connection id for an open
                               // SMS gateway connection
    boolean:  markAsRead      // Flag to instruct the directSMS
                               // gateway to mark the retrieved
                               // messages as "read"
}

```

The sample message shown below, will retrieve all replies to 2-Way SMS messages that are in the "unread" state.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:getAllSmsMessageReplies>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
      <markAsRead xsi:type="xsd:boolean">false</markAsRead>
    </ns4:getAllSmsMessageReplies>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message is slightly more complicated in that it contains an array of SmsReply objects. Each one of these objects contains the message text, the mobile number of the originator of the message, along with any "message id" passed when the original 2-Way SMS message was submitted as well as the number of seconds ago this reply message was received.

```

/*
 * Data structure to model the reply to a 2-way SMS message
 */
SmsReply
{
    String: message    // Message text
    String: messageId // Message id passed in the original 2-way
                       // message sent earlier
    String: mobile     // mobile number of the message sender
    long:  when        // How long ago this message was received
}

```

```

        // in seconds
    }

    /*
     * Successful response to an getAllSmsReplies operation, it contains
     * a collection of all "unread" reply messages found
     */
    GetAllSmsMessageRepliesResponse
    {
        SmsReply[]: replies // Array of SMS response messages
    }

```

The sample message below shows 4 SMS response message retrieved.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <ns1:getAllSmsMessageRepliesResponse
            soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:ns1="urn:SmsGateway">
            <getAllSmsMessageRepliesReturn
                xsi:type="soapenc:Array"
                soapenc:arrayType="ns1:SmsReply[5]"
                xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
                <item href="#id0"/>
                <item href="#id1"/>
                <item href="#id2"/>
                <item href="#id3"/>
                <item href="#id4"/>
            </getAllSmsMessageRepliesReturn>
        </ns1:getAllSmsMessageRepliesResponse>
        <multiRef id="id3"
            soapenc:root="0"
            soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xsi:type="ns2:SmsReply"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:ns2="urn:SmsGateway">
            <message xsi:type="xsd:string">
                don't bug ME
            </message>
            <messageId xsi:type="xsd:string">my-id</messageId>
            <mobile xsi:type="xsd:string">+61 414 574 496</mobile>
            <when xsi:type="xsd:long">1788186</when>
        </multiRef>
        <multiRef id="id0"
            soapenc:root="0"
            soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xsi:type="ns3:SmsReply"
            xmlns:ns3="urn:SmsGateway"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
            <message xsi:type="xsd:string">
                I am sending this message to test replies. Aagain.
            </message>
            <messageId xsi:type="xsd:string">my_message_id_1</messageId>
            <mobile xsi:type="xsd:string">+61 407 263 977</mobile>
            <when xsi:type="xsd:long">19881017</when>
        </multiRef>
        <multiRef id="id1"
            soapenc:root="0"
            soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xsi:type="ns4:SmsReply"
            xmlns:ns4="urn:SmsGateway"
            xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
            <message xsi:type="xsd:string">
                This should be reply number three.
            </message>
            <messageId xsi:type="xsd:string">my_message_id_1</messageId>
            <mobile xsi:type="xsd:string">+61 407 263 977</mobile>

```

```

    <when xsi:type="xsd:long">19808710</when>
  </multiRef>
  <multiRef id="id4"
    soapenc:root="0"
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    xsi:type="ns5:SmsReply"
    xmlns:ns5="urn:SmsGateway"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    <message xsi:type="xsd:string">
      this is a test
    </message>
    <messageId xsi:type="xsd:string">my_message_id_1</messageId>
    <mobile xsi:type="xsd:string">+61 411 206 954</mobile>
    <when xsi:type="xsd:long">1201</when>
  </multiRef>
  <multiRef id="id2"
    soapenc:root="0"
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    xsi:type="ns6:SmsReply"
    xmlns:ns6="urn:SmsGateway"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    <message xsi:type="xsd:string">
      I am sending this message to test replies.
    </message>
    <messageId xsi:type="xsd:string">my_message_id_1</messageId>
    <mobile xsi:type="xsd:string">+61 407 263 977</mobile>
    <when xsi:type="xsd:long">19881111</when>
  </multiRef>
</soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the connection id used in the request belonged to gateway connection that was already closed.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        connection already closed
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```


getSmsMessageReplies ()

This operation is very similar to **geAllSmsReplies** except that it searches for replies to a specific 2-way message.

SOAP Request

The request consists of the message id of a 2-way message submitted earlier and a boolean flag to indicate to the directSMS gateway whether or not you want to mark the reply messages returned as "read". This stops them from being retrieved in future calls to this operation as well as the **getAllSmsReplies** operation.

```

/*
 * Request to retrieve all SMS responses to a specific 2-way SMS
 * message
 */
GetSmsMessageRepliesRequest
{
    String:  connectionId    // Connection id for an open connection
    String:  messageId      // Message id corresponding to the one
                          // associated with a 2-way SMS
    boolean: markAsRead     // Mark the replies retrieved as "read"
                          // and so will not be retrieved again
                          // in future calls to this operation
}

```

In the example below we are looking for all "unread" replies to the 2-way SMS with message id = 'my_message_id_1'. And we are not going to mark those replies as read, which means they are going to be fetched again in future queries.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:getSmsMessageReplies>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
      <messageId xsi:type="xsd:string">my_message_id_1</messageId>
      <markAsRead xsi:type="xsd:boolean">false</markAsRead>
    </ns4:getSmsMessageReplies>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message is slightly more complicated in that it contains an array of SmsReply objects. Each one of these objects contains the message text, the mobile number of the originator of the message, along with the "message id" you passed when the original 2-way message was submitted and the number of seconds this message was received.

```

/*
 * Data structure to model the response to a 2-way SMS message
 */
SmsReply

```

```

{
    String: message      // Message text
    String: messageId   // Message id passed in the original 2-way
                        // message sent earlier
    String: mobile      // mobile number of the message sender
    long:  when         // How long ago this message was received
                        // in seconds
}

/*
 * Successful response to an getAllSmsReplies operation, it contains
 * a collection of all "unread" reply messages found
 */
GetAllSmsMessageRepliesResponse
{
    SmsReply[]: replies
}

```

Note that mobile numbers are returned in the format of +61 xxx yyy zzz in order to not confuse some non typed languages such as PHP which would try to convert the String "61423123123" to an integer and thus give clients the wrong information. Hence the addition of the '+' character and the spaces.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getSmsMessageRepliesResponse soapenv:encodingStyle="
      http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <getSmsMessageRepliesReturn
        xsi:type="soapenc:Array"
        soapenc:arrayType="ns1:SmsReply[4]"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <item href="#id0"/>
        <item href="#id1"/>
        <item href="#id2"/>
        <item href="#id3"/>
      </getSmsMessageRepliesReturn>
    </ns1:getSmsMessageRepliesResponse>
    <multiRef id="id2"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:SmsReply"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns2="urn:SmsGateway">
      <message xsi:type="xsd:string">
        I am sending this message to test replies.
      </message>
      <messageId xsi:type="xsd:string">my_message_id_1</messageId>
      <mobile xsi:type="xsd:string">+61 407 263 977</mobile>
      <when xsi:type="xsd:long">19881111</when>
    </multiRef>
    <multiRef id="id3"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:SmsReply"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns2="urn:SmsGateway">
      <message xsi:type="xsd:string">
        this is a test
      </message>
      <messageId xsi:type="xsd:string">my_message_id_1</messageId>
      <mobile xsi:type="xsd:string">+61 411 206 954</mobile>
      <when xsi:type="xsd:long">1201</when>
    </multiRef>
    <multiRef id="id1"

```

```

        soapenc:root="0"
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xsi:type="ns2:SmsReply"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns2="urn:SmsGateway">
        <message xsi:type="xsd:string">
            This should be reply number three.
        </message>
        <messageId xsi:type="xsd:string">my_message_id_1</messageId>
        <mobile xsi:type="xsd:string">+61 407 263 977</mobile>
        <when xsi:type="xsd:long">19808710</when>
    </multiRef>
    <multiRef id="id0"
        soapenc:root="0"
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xsi:type="ns2:SmsReply"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns2="urn:SmsGateway">
        <message xsi:type="xsd:string">
            I am sending this message to test replies. And again.
        </message>
        <messageId xsi:type="xsd:string">my_message_id_1</messageId>
        <mobile xsi:type="xsd:string">+61 407 263 977</mobile>
        <when xsi:type="xsd:long">19881017</when>
    </multiRef>

</soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the connection id used in the request belonged to gateway connection that was already closed.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>
                soapenv:Server.userException
            </faultcode>
            <faultstring>
                com.directsms.s3.service.sms.SmsServiceException:
                connection already closed
            </faultstring>
            <detail/>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>

```

getAllInboundSmsMessages ()

The operation to fetch all "unread" inbound SMS messages.

SOAP Request

The request contains a boolean flag to indicate to the directSMS gateway whether or not you want to mark the messages returned as "read". This stops them from being retrieved in future calls to this operation as well as *getInboundSmsMessages*.

```

/*
 * Request to retrieve all responses to 2-way SMS messages sent
 */
GetAllInboundSmsMessagesRequest
{
    String:   connectionId      // Connection id for an open
                                // SMS gateway connection
    boolean:  markAsRead        // Flag to instruct the directSMS
                                // gateway to mark the retrieved
                                // messages as "read"
}

```

The sample message shown below will retrieve all inbound SMS messages that are in the "unread" state.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:getAllInboundSmsMessages>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
      <markAsRead xsi:type="xsd:boolean">false</markAsRead>
    </ns4:getAllInboundSmsMessages>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message is slightly more complicated in that it contains an array of InboundSmsMessage objects. Each one of these objects contains the message text, the mobile number of the message sender, the inbound number the message was received on as well as the number of seconds ago this message was received.

```

/*
 * Data structure to model an inbound SMS message
 */
InboundSmsMessage
{
    String:  message           // Message text
    String:  inboundNumner     // The number this message was received on
    String:  mobile           // Mobile number of the message sender
    long:    when              // How long ago this message was received
                                // in seconds
}

```

```

}

/*
 * Successful response to an getAllInboundSmsMessages operation,
 * it contains a collection of all "unread" messages found
 */
GetAllInboundSmsMessagesResponse
{
    InboundSmsMessage[]: messages    // Array of inbound SMS messages
}

```

The sample message below shows 2 SMS response message retrieved.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getAllInboundSmsMessagesResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <getAllInboundSmsMessagesReturn
        xsi:type="soapenc:Array"
        soapenc:arrayType="ns1:InboundSmsMessage[2]"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <item href="#id0"/>
        <item href="#id1"/>
      </getAllInboundSmsMessagesReturn>
    </ns1:getAllInboundSmsMessagesResponse>
    <multiRef id="id0"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:InboundSmsMessage"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns2="urn:SmsGateway">
      <message xsi:type="xsd:string">
        This is a test of the inbound solution
      </message>
      <inboundNumber xsi:type="xsd:string">+61 411 111 111</inboundNumber>
      <mobile xsi:type="xsd:string">+61 414 574 496</mobile>
      <when xsi:type="xsd:long">1788186</when>
    </multiRef>
    <multiRef id="id1"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns3:InboundSmsMessage"
      xmlns:ns3="urn:SmsGateway"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
      <message xsi:type="xsd:string">
        I am sending this message to test inbound messaging.
      </message>
      <inboundNumber xsi:type="xsd:string">+61 411 111 111</inboundNumber>
      <mobile xsi:type="xsd:string">+61 407 263 977</mobile>
      <when xsi:type="xsd:long">19881017</when>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the connection id used in the request belonged to gateway connection that was already closed.

```

<?xml version="1.0" encoding="UTF-8"?>

```

directSMS SOAP API

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        connection already closed
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

getInboundSmsMessages ()

The operation to fetch all "unread" inbound SMS messages received on the specified inbound number.

SOAP Request

The request contains the inbound number you wish to query for new inbound messages, as well as a boolean flag to indicate to the directSMS gateway whether or not you want to mark the messages returned as "read". This stops them from being retrieved in future calls to this operation as well as **getAllInboundSmsMessages**.

```

/*
 * Request to retrieve all inbound SMS messages received
 */
GetInboundSmsMessagesRequest
{
    String:  connectionId      // Connection id for an open
                                // SMS gateway connection
    String:  inboundNumber    // Inbound number that received the
                                // messages
    boolean: markAsRead       // Flag to instruct the directSMS
                                // gateway to mark the retrieved
                                // messages as "read"
}

```

The sample message shown below will retrieve all inbound SMS messages that are in the "unread" state received on 0411 111 111.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:getInboundSmsMessages>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
      <inboundNumber xsi:type="xsd:string">
        0411 111 111
      </inboundNumber>
      <markAsRead xsi:type="xsd:boolean">false</markAsRead>
    </ns4:getAllInboundSmsMessages>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response message is slightly more complicated in that it contains an array of InboundSmsMessage objects. Each one of these objects contains the message text, the mobile number of the message sender, the inbound number the message was received on as well as the number of seconds ago this message was received.

```

/*
 * Data structure to model an inbound SMS message
 */

```

```

InboundSmsMessage
{
    String: message           // Message text
    String: inboundNumner    // The number this message was received on
    String: mobile           // Mobile number of the message sender
    long:   when              // How long ago this message was received
                                // in seconds
}

/*
 * Successful response to an getInboundSmsMessages operation,
 * it contains a collection of all "unread" messages found
 */
GetInboundSmsMessagesResponse
{
    InboundSmsMessage[]: messages // Array of inbound SMS messages
}

```

The sample message below shows 2 SMS response message retrieved.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getInboundSmsMessagesResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <getInboundSmsMessagesReturn
        xsi:type="soapenc:Array"
        soapenc:arrayType="ns1:InboundSmsMessage[2]"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <item href="#id0"/>
        <item href="#id1"/>
      </getInboundSmsMessagesReturn>
    </ns1:getInboundSmsMessagesResponse>
    <multiRef id="id0"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns2:InboundSmsMessage"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns2="urn:SmsGateway">
      <message xsi:type="xsd:string">
        This is a test of the inbound solution
      </message>
      <inboundNumber xsi:type="xsd:string">+61 411 111 111</inboundNumber>
      <mobile xsi:type="xsd:string">+61 414 574 496</mobile>
      <when xsi:type="xsd:long">1788186</when>
    </multiRef>
    <multiRef id="id1"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="ns3:InboundSmsMessage"
      xmlns:ns3="urn:SmsGateway"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
      <message xsi:type="xsd:string">
        I am sending this message to test inbound messaging.
      </message>
      <inboundNumber xsi:type="xsd:string">+61 411 111 111</inboundNumber>
      <mobile xsi:type="xsd:string">+61 407 263 977</mobile>
      <when xsi:type="xsd:long">19881017</when>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>

```


SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the connection id used in the request belonged to gateway connection that was already closed.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        connection already closed
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

getCurrentBalance ()

Operation to get the customer's remaining credit balance.

SOAP Request

This is an empty request message triggering the gateway to retrieve the SMS credit balance for the currently connected user.

```

/*
 * Request for the current directSMS credit balance
 */
GetCurrentBalanceRequest
{
    String: connectionId      // Connection id for an open
                             // SMS gateway connection
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:getCurrentBalance>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
    </ns4:getCurrentBalance>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response is the number of directSMS' credits left on the customer account.

```

/*
 * The successful response to a credit balance request
 */
GetCurrentBalanceResponse
{
    double: balance          // Number of directSMS credits remaining
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getCurrentBalanceResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="urn:SmsGateway">
      <getCurrentBalanceReturn xsi:type="xsd:double">
        11286.0
      </getCurrentBalanceReturn>
    </ns1:getCurrentBalanceResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the connection id used in the request belonged to gateway connection that was already closed.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        connection already closed
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

disconnect ()

Once all operations are carried out and the customer no longer needs the connection. This operation is called in order to close the connection and perform some clean up tasks on the server. **All inactive connections will be closed automatically after a period of 6 hours.**

SOAP Request

The request message is empty. It triggers the SMS gateway to disconnect the currently connected user.

```

/*
 * Request to disconnect from the server and shut down the
 * gateway connection
 */
DisconnectRequest
{
    String: connectionId      // Connection id for an open
                             // SMS gateway connection
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:ns4="urn:SmsGateway"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns4:disconnect>
      <connectionId xsi:type="xsd:string">
        f4b9ec30ad9f68f89b29639786cb62ef
      </connectionId>
    </ns4:disconnect>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SOAP Response

The response will contain the connection id submitted in the request. Which signifies that the gateway connection has been closed.

```

/*
 * Respond with whether or not the gateway connection was closed
 * successfully
 */
DisconnectResponse
{
    boolean:      connectionClosed // Set to true if the connection
                             // was closed
}

```

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:disconnectResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

```

```
xmlns:ns1="urn:SmsGateway">
  <disconnectReturn xsi:type="xsd:boolean">
    true
  </disconnectReturn>
</ns1:disconnectResponse>

</soapenv:Body>
</soapenv:Envelope>
```

SOAP Fault

The SOAP fault object is returned in the event of any errors or exception conditions encountered. The fault returned will contain a faultstring element which should tell you exactly what went wrong. In the example below, the connection id used in the request belonged to gateway connection that was already closed.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>
        soapenv:Server.userException
      </faultcode>
      <faultstring>
        com.directsms.s3.service.sms.SmsServiceException:
        connection already closed
      </faultstring>
      <detail/>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```