



Gateway Web Service Specification Document

COPYRIGHT NOTICE

The material contained on the following pages, including but not limited to all trademarks (whether registered or not), inventions, patents (both registered and unregistered), registered and unregistered designs, know-how and other intellectual property contained and/or noted therein ("the material"), shall remain the sole property of Blue Label Telecoms Ltd (BLT). The material is subject to copyright protection under the Convention for the Protection of Literary and Artistic Works of 1886 ("Berne Convention") and the Agreement on the Trade-Related Aspects of Intellectual Property Rights. No part of the material may be reproduced or transmitted in any form whatsoever, whether by any electronic or mechanical means, including photocopying, or by any information storage and retrieval system, without the prior and express written permission of BLT. Any unauthorized reproduction or transmission of the material will constitute a copyright infringement and render the doer liable under both civil and criminal law.

Document Status	RELEASED
Version:	1.5
Filename:	Gateway Web Service Specification Document.docx
Last Revision Date:	Tuesday, 29 October 2013
Reviewer(s)	

TABLE OF CONTENTS

1	INTRODUCTION	4
2	WEB SERVICE PUBLIC METHODS OVERVIEW	4
2.1	LOGIN	4
2.1.1	SOAP	4
2.1.2	HTTP GET	4
2.2	LOGOUT	5
2.2.1	SOAP	5
2.2.2	HTTP GET	5
2.3	GETSESSIONSTATUS	5
2.3.1	SOAP	5
2.3.2	HTTP GET	5
2.4	GETCREDITS	6
2.4.1	SOAP	6
2.4.2	HTTP GET	6
2.5	BULKSMSFILEUPLOAD	6
2.6	BULKSMSFILEUPLOADSCHEDULED	7
2.7	GETINCOMINGMESSAGEDETAILBYID	7
2.7.1	SOAP	7
2.7.2	HTTP GET	8
2.8	GETINCOMINGMESSAGEIDSBYUSERDATASET	9
2.8.1	SOAP	9
2.8.2	HTTP GET	9
2.9	GETINCOMINGMESSAGESBYSHORTCODE	10
2.9.1	SOAP	10
2.9.2	HTTP GET	10
2.10	GETINCOMINGMESSAGESBYUSER	11
2.10.1	SOAP	11
2.10.2	HTTP GET	11
2.11	GETINCOMINGMESSAGESBYUSERDATASET	12
2.11.1	SOAP	12
2.11.2	HTTP GET	12
2.12	GETINCOMINGMESSAGESFORDATERANGE	12
2.12.1	SOAP	12
2.12.2	HTTP GET	13
2.13	GETSENTSMSMESSAGEDETAILBYID	13
2.13.1	SOAP	13
2.13.2	HTTP GET	13
2.14	SCHEDULESMSMESSAGESINGLE	15
2.14.1	SOAP	15
2.14.2	HTTP GET	15
2.15	SENDSMSMESSAGESINGLE	16
2.15.1	SOAP	16
2.15.2	HTTP GET	16
2.16	SENDWAPPUSHSINGLE	16
2.16.1	SOAP	16
2.16.2	HTTP GET	17

2.17	GETINCOMINGMESSAGESBYSHORTCODEONCE.....	17
2.17.1	<i>SOAP</i>	17
2.17.2	<i>HTTP GET</i>	17
2.18	GETINCOMINGMESSAGESBYSHORTCODEONCEADMIN	18
2.18.1	<i>SOAP</i>	18
2.18.2	<i>HTTP GET</i>	19
2.19	SENDOTASETTINGS	21
2.19.1	<i>SOAP</i>	21
2.20	SENDMMSSINGLEUPLOADFILES.....	22
2.21	SENDSINGLEHLRLOOKUPREQUEST	24
3	PUSHSERVICESPECIFICATIONS	27
4	POSSIBLE ERROR CODES RETURNED FROM THE WEB SERVICE	29
5	AVAILABLE NUMBER FORMATS.....	29
6	AVAILABLE GATEWAY WEB SERVICE FUNCTIONS	30

1 Introduction

Cellfind has developed a Web service for submitting and retrieving SMS messages to and from the Cellfind Messaging Gateway. This document will show and describe the core functionality that is available for use. The Web service can be accessed directly on any SOAP enabled programming language. Microsoft.NET C# was used for the example code.

The Web service can be accessed from:

- <https://www.cellportal.co.za/gatewaywebservice/Service.asmx>
- WSDL Definition
<https://www.cellportal.co.za/gatewaywebservice/Service.asmx?WSDL>

2 Web Service Public methods overview

2.1 LOGIN

Log the user into the system and get a session token back.

2.1.1 SOAP

Code Snippet:

```
string sessionToken = _service.Login(userName, password);
```

Parameters:

- userName (string) : A valid username as supplied by CellFind / the system.
- password (string) : A valid password as supplied by CellFind / the system.

Returns : A session token string (SessionToken).

2.1.2 HTTP GET

Code Snippet:

```
https://www.cellportal.co.za/gatewaywebservice/Service.asmx/Login?username=userName&password=password
```

Parameters:

- userName (string) : A valid username as supplied by CellFind / the system.
- password (string) : A valid password as supplied by CellFind / the system.

Returns : A session token string (SessionToken) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>  
<string xmlns="http://tempuri.org/">123456789</string>
```

2.2 LOGOUT

Log the user out of the system and terminate the current session.

2.2.1 SOAP

Code Snippet:

```
string sessionStatus = _service.LogOut(sessionToken);
```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service

Returns : A session token string (SessionToken).

2.2.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/LogOut?session_token=123456789

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service

Returns : A session token string (SessionToken) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>  
<string xmlns="http://tempuri.org/">SessionTerminated</string>
```

2.3 GETSESSIONSTATUS

Get the session status for a specific session.

2.3.1 SOAP

Code Snippet:

```
string SessionStatus = _service.GetSessionStatus(sessionToken);
```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service

Returns : A status string (SessionActive,SessionExpired,InvalidSessionToken)

2.3.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetSessionStatus?session_token=123456789

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service

Returns : A status string (SessionActive,SessionExpired,InvalidSessionToken) in XML format

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

```
<string xmlns="http://tempuri.org/">SessionExpired</string>
```

2.4 GETCREDITS

Return the amount of credits available.

2.4.1 SOAP

Code Snippet:

```
int credits = _service.GetCredits(sessionToken);
```

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : An int value with the amount of credits available.

2.4.2 HTTP GET

Code Snippet:

```
http://designdesktop/gatewaywebservice/Service.asmx/GetCredits?session_token=123456789
```

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : An int value with the amount of credits available.

```
<?xml version="1.0" encoding="utf-8"?>  
<int xmlns="http://tempuri.org/">210</int>
```

2.5 BULKSMSFILEUPLOAD

Upload CSV files to the web server for bulk sms messages.

File example:

Recipient	Reference	Message
0828092776	Ref001	Message text for message No 1

Code Snippet:

```
FileInfo fileInfo = new FileInfo("c:/ExampleFile.csv");  
string filename = fileInfo.Name;  
int numBytes = fileInfo.Length;  
  
FileStream fStream = new FileStream(filename, FileMode.Open, FileAccess.Read);  
BinaryReader br = new BinaryReader(fStream);  
byte[] fileData = br.ReadBytes(numBytes);  
br.Close();  
string result = _service.BulkSMSFileUpload(fileData, fileName, sessionToken);  
fStream.Close();  
fStream.Dispose();
```

Parameters:

- fileData : The byte array data for the file.
- fileName : The original filename that is being uploaded.

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service

Returns : The status of the upload. (OK if successful)

2.6 BULKSMSFILEUPLOADSCHEDULED

Upload a CSV file to the web server for bulk sms messages that will be scheduled.

File example:

Recipient	Reference	Message
0828092776	Ref001	Message text for message No 1

Code Snippet:

```
String strFile = System.IO.Path.GetFileName("c:/ExampleFile.csv");
FileStream fStream = new FileStream(filename, FileMode.Open, FileAccess.Read);
BinaryReader br = new BinaryReader(fStream);
byte[] data = br.ReadBytes((int)numBytes);
br.Close();
string sTmp = _service.BulkSMSFileUploadScheduled(data, strFile, sessionToken, ScheduledDateTime);
fStream.Close();
fStream.Dispose();
```

Paramaters:

- fileData : The byte array data for the file.
- fileName : The original filename that is being uploaded.
- sessionToken : A valid logged in sessionToken as generated by the Gateway web service
- dateTime (DateTime) : A valid date/time value for the date/time that the message needs to be scheduled for.

Returns : The status of the upload. (OK if successful)

2.7 GETINCOMINGMESSAGEDETAILBYID

Get the detail for a specific incoming message by ID.

2.7.1 SOAP

Code Snippet:

```
dataSet = _service.GetIncomingMessageDetailByID(sessionToken, messageID);
```

Paramaters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.
- messageID : A valid message ID of a message that has been sent.

Returns : A dataset with the detail for the specific message. (Message_ID, Batch_ID, Priority_ID, Reference, Receive_Time, Schedule_Time, Process_Time, Recipient_No, Subject, Message, Protocol_ID, Status_ID, User_ID, Tag, Submit_Time, Sent_Time, Bill_Date, Network, Application, qid)

2.7.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessageDetailById?session_token=123456789&reply_message_id=123456

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.
- messageID : A valid message ID of a message that has been sent.

Returns : A dataset with the detail for the specific message. (Message_ID, Batch_ID, Priority_ID, Reference, Receive_Time, Schedule_Time, Process_Time, Recipient_No, Subject, Message, Protocol_ID, Status_ID, User_ID, Tag, Submit_Time, Sent_Time, Bill_Date, Network, Application, qid) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://tempuri.org/">
  <xs:schema id="ReturnDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="ReturnDataSet" msdata:lsDataSet="true" msdata:UseCurrentLocale="true">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="ReplyDataTable">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Reply_Message_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Destination_No" type="xs:string" minOccurs="0" />
                <xs:element name="Message_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Message" type="xs:string" minOccurs="0" />
                <xs:element name="Originator" type="xs:string" minOccurs="0" />
                <xs:element name="Protocol_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Receive_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Reported" type="xs:string" minOccurs="0" />
                <xs:element name="Report_Date" type="xs:string" minOccurs="0" />
                <xs:element name="Network" type="xs:string" minOccurs="0" />
                <xs:element name="Tag" type="xs:string" minOccurs="0" />
                <xs:element name="User_ID" type="xs:string" minOccurs="0" />
                <xs:element name="smsid" type="xs:string" minOccurs="0" />
                <xs:element name="Push_Date" type="xs:string" minOccurs="0" />
                <xs:element name="Pushed" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
  <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
    <ReturnDataSet xmlns="">
      <ReplyDataTable diffgr:id="ReplyDataTable1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
        <Reply_Message_ID>108709</Reply_Message_ID>
        <Destination_No>27839300217</Destination_No>
        <Message_ID>1240183</Message_ID>
        <Message>Mgd</Message>
        <Originator>27784567890</Originator>
        <Protocol_ID>1</Protocol_ID>
        <Receive_Time>8/19/2009 12:41:18 PM</Receive_Time>
        <Reported>0</Reported>
        <Network>2</Network>
        <Tag>12345</Tag>
        <User_ID>1</User_ID>
        <smsid>1234567</smsid>
      </ReplyDataTable>
    </ReturnDataSet>
  </diffgr:diffgram>

```


</DataSet>

2.8 GETINCOMINGMESSAGEIDSBYUSERDATASET

Get a list of incoming SMS message IDs for a user - returns a dataset

2.8.1 SOAP

Code Snippet:

```
dataSet = _service.GetIncomingMessageIdsByUserDataSet(txtToken.Text.ToString());
```

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the message IDs (Reply_Message_ID).

2.8.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessageIdsByUserDataSet?session_token=123456789

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the message IDs (Reply_Message_ID) in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://tempuri.org/">
  <xs:schema id="ReturnDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="ReturnDataSet" msdata:isDataSet="true" msdata:UseCurrentLocale="true">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="ReplyDataTable">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Reply_Message_ID" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-
microsoft-com:xml-diffgram-v1">
  <ReturnDataSet xmlns="">
    <ReplyDataTable diffgr:id="ReplyDataTable1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
      <Reply_Message_ID>123456</Reply_Message_ID>
    </ReplyDataTable>
    <ReplyDataTable diffgr:id="ReplyDataTable2" msdata:rowOrder="1" diffgr:hasChanges="inserted">
      <Reply_Message_ID>234567</Reply_Message_ID>
    </ReplyDataTable>
    <ReplyDataTable diffgr:id="ReplyDataTable3" msdata:rowOrder="2" diffgr:hasChanges="inserted">
      <Reply_Message_ID>345678</Reply_Message_ID>
    </ReplyDataTable>
  </ReturnDataSet>
</diffgr:diffgram>
</DataSet>
```

2.9 GETINCOMINGMESSAGESBYSHORTCODE

Get a list of incoming SMS messages by short code

2.9.1 SOAP

Code Snippet:

```
dataSet = _service.GetIncomingMessagesByShortCode(shortCode,sessionToken);
```

Parameters:

- shortCode : A valid short code number to get the received messages again.
- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid).

2.9.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessagesByShortCode?session_token=123456789&short_code=31100

Parameters:

- shortCode : A valid short code number to get the received messages again.
- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://tempuri.org/">
  <xs:schema id="ReturnDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="ReturnDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="ReplyDataTable">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Reply_Message_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Destination_No" type="xs:string" minOccurs="0" />
                <xs:element name="Message_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Message" type="xs:string" minOccurs="0" />
                <xs:element name="Originator" type="xs:string" minOccurs="0" />
                <xs:element name="Protocol_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Receive_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Reported" type="xs:string" minOccurs="0" />
                <xs:element name="Report_Date" type="xs:string" minOccurs="0" />
                <xs:element name="Network" type="xs:string" minOccurs="0" />
                <xs:element name="Tag" type="xs:string" minOccurs="0" />
                <xs:element name="User_ID" type="xs:string" minOccurs="0" />
                <xs:element name="smsid" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>

```

```
<xs:element name="Push_Date" type="xs:string" minOccurs="0" />
<xs:element name="Pushed" type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
  <ReturnDataSet xmlns="">
    <ReplyDataTable diffgr:id="ReplyDataTable1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
      <Reply_Message_ID>123</Reply_Message_ID>
      <Destination_No>31100</Destination_No>
      <Message_ID>0</Message_ID>
      <Message>Stop</Message>
      <Originator>0821234567</Originator>
      <Protocol_ID>1</Protocol_ID>
      <Receive_Time>8/11/2008 10:00:51 PM</Receive_Time>
      <Reported>0</Reported>
      <Network>1</Network>
      <Tag />
      <User_ID>1</User_ID>
      <smsid>1234567</smsid>
    </ReplyDataTable>
  </ReturnDataSet>
</diffgr:diffgram>
</DataSet>
```

2.10 GETINCOMINGMESSAGESBYUSER

Get a list of incoming SMS messages by user

2.10.1 SOAP

Code Snippets:

```
string test = _service.GetIncomingMessagesByUser(sessionToken);
```

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid).

2.10.2 HTTP GET

Code Snippets:

```
https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessagesByUser?session\_token=123456789
```

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid) in XML format.

2.11 GETINCOMINGMESSAGESBYUSERDATASET

Get a list of incoming SMS messages by user - returns a dataset

2.11.1 SOAP

Code Snippet:

```
dataSet = _service.GetIncomingMessagesByUserDataSet(sessionToken);
```

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid).

2.11.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessagesByUserDataSet?session_token=123456789

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid) in XML format.

2.12 GETINCOMINGMESSAGESFORDATERANGE

Get a list of incoming SMS messages by User for a date range - returns a dataset

2.12.1 SOAP

Code Snippet:

```
System.Data.DataSet ds  
= _service.GetIncomingMessagesForDateRange(sessionToken, fromDate, toDate);
```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- fromDate (DateTime) : The start date for the report.
- toDate (DateTime) : The end data for the report.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid).

2.12.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessagesForDateRange?session_token=1615091861&from_date=2009-01-01&to_Date=2009-08-31&reported=false

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- fromDate (DateTime) : The start date for the report.
- toDate (DateTime) : The end data for the report.
- reported (Boolean) :
 - True – Includes messages that have been reported on
 - False – Excludes messages that have been reported on

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid) in XML format.

2.13 GETSENTSMSMESSAGEDETAILBYID

Get the detail for sent SMS messages by Message ID- returns a dataset

2.13.1 SOAP

Code Snippet:

```
System.Data.DataSet ds = _service.GetSentSMSMessageDetailByID(sessionToken, messageID);
```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- messageID (int) : A valid ID for the message that was submitted.

Returns : A dataset with the sent message detail (Message_ID, Batch_ID, Priority_ID, Reference, Receive_Time, Schedule_Time, Process_Time, Recipient_No, Subject, Message, Protocol_ID, Status_ID, User_ID, Tag, Submit_Time, Sent_Time, Bill_Date, Network, Application, qid).

2.13.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetSentSMSMessageDetailByID?session_token=123456789&message_id=1

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- messageID (int) : A valid ID for the message that was submitted.

Returns : A dataset with the sent message detail (Message_ID, Batch_ID, Priority_ID, Reference, Receive_Time, Schedule_Time, Process_Time, Recipient_No, Subject, Message, Protocol_ID, Status_ID, User_ID, Tag, Submit_Time, Sent_Time, Bill_Date, Network, Application, qid) in XML format.

```

<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://tempuri.org/">
  <xs:schema id="ReturnDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="ReturnDataSet" msdata:isDataSet="true" msdata:UseCurrentLocale="true">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="SentSMSDataTable">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Message_ID" type="xs:string" />
                <xs:element name="Batch_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Priority_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Reference" type="xs:string" minOccurs="0" />
                <xs:element name="Receive_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Schedule_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Process_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Recipient_No" type="xs:string" minOccurs="0" />
                <xs:element name="Subject" type="xs:string" minOccurs="0" />
                <xs:element name="Message" type="xs:string" minOccurs="0" />
                <xs:element name="Protocol_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Status_ID" type="xs:string" minOccurs="0" />
                <xs:element name="User_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Tag" type="xs:string" minOccurs="0" />
                <xs:element name="Submit_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Sent_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Bill_Date" type="xs:string" minOccurs="0" />
                <xs:element name="Network" type="xs:string" minOccurs="0" />
                <xs:element name="Application" type="xs:string" minOccurs="0" />
                <xs:element name="qid" type="xs:string" minOccurs="0" />
                <xs:element name="Auto_Reply" type="xs:string" minOccurs="0" />
                <xs:element name="Message_Type" type="xs:string" minOccurs="0" />
                <xs:element name="Delivery_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Message_Queue_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Bulk_File_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Queue" type="xs:string" minOccurs="0" />
                <xs:element name="Credits_Used" type="xs:string" minOccurs="0" />
                <xs:element name="Credit_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Message_Status" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
  <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
    <ReturnDataSet xmlns="">
      <SentSMSDataTable diffgr:id="SentSMSDataTable1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
        <Message_ID>1</Message_ID>
        <Batch_ID>0</Batch_ID>
        <Priority_ID>2</Priority_ID>
        <Reference />
        <Receive_Time>6/26/2008 11:34:17 AM</Receive_Time>
        <Schedule_Time>1/1/0001 12:00:00 AM</Schedule_Time>
        <Process_Time>6/26/2008 1:06:50 PM</Process_Time>
        <Recipient_No>0821234567</Recipient_No>
        <Subject />
        <Message>Test message from the gateway.</Message>
        <Protocol_ID>1</Protocol_ID>
        <Status_ID>8</Status_ID>
        <User_ID>1</User_ID>
        <Tag>10002</Tag>
        <Submit_Time>6/26/2008 11:34:17 AM</Submit_Time>
        <Sent_Time>6/26/2008 1:06:50 PM</Sent_Time>
        <Network>1</Network>
        <qid>1234567</qid>
        <Auto_Reply>N</Auto_Reply>
        <Message_Type>1</Message_Type>
        <Delivery_Time>6/26/2008 1:07:50 PM</Delivery_Time>
        <Bulk_File_ID>0</Bulk_File_ID>
      </SentSMSDataTable>
    </ReturnDataSet>
  </diffgr:diffgram>

```

```

<Queue>1</Queue>
<Credits_Used>0</Credits_Used>
<Credit_ID>0</Credit_ID>
<Message_Status>delivered</Message_Status>
</SentSMSDataTable>
</ReturnDataSet>
</diffgr:diffgram>
</DataSet>

```

Message Status :	1	received
	2	submitted
	4	sent
	5	failed
	3	queued
	8	delivered
	9	expired
	7	error
	6	retry
	11	exclusion
	12	invalid number
	13	call limit reached
	14	duplicate

2.14 SCHEDULESMSMESSAGESINGLE

Schedule a single SMS message using a session token.

2.14.1 SOAP

Code Snippet:

```

int messageID = _service.ScheduleSMSMessageSingleST(sessionToken, recipient,
message,dateTime);

```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- recipient (string) : Valid recipient for the message.
- message (string) : The 160 character message to send to the recipient.
- dateTime (DateTime) : A valid date/time value for the date/time that the message needs to be scheduled for.

Returns : The message ID of the newly created message (value).

2.14.2 HTTP GET

Code Snippet:

```

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/ScheduleSMSMessageSingleHttpGet?session_token=123456789&mobile=27721234567&message=this+is+a+test+message+)(*%26^%25%24%23%40![]{}%3B"
<>.%2C&schedule_time=2009-08-21+13%3A00%3A00

```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- recipient (string) : Valid recipient for the message.
- message (string) : The 160 character message to send to the recipient. Please note that the **message** has to be **URL ENCODED**.
- dateTime (DateTime) : A valid date/time value for the date/time that the message needs to be scheduled for. Please note that the **dateTime** has to be **URL ENCODED**.

Returns : The message ID of the newly created message (value) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<int xmlns="http://tempuri.org/">1250924</int>
```

2.15 SENDSMSMESSAGESINGLE

Send a single SMS message using a session token.

2.15.1 SOAP

Code Snippet:

```
int MessageID = _service.SendSMSMessageSingleST(sessionToken, recipient,message);
```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- recipient (string) : A valid recipient number.
- message (string) : The 160 character message to send to the recipient.

Returns: The message ID of the newly created message (value).

2.15.2 HTTP GET

Code Snippet:

```
https://www.cellportal.co.za/gatewaywebservice/Service.asmx/SendSMSMessageSingleHttpGet?session_token=123456789&mobile=27721234567&message=this+is+a+test+message+)(*%26^%25%24%23%40![]}%3B"<>.%2C
```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- recipient (string) : A valid recipient number.
- message (string) : The 160 character message to send to the recipient Please note that the message has to be **URL ENCODED**.

Returns: The message ID of the newly created message (value) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<int xmlns="http://tempuri.org/">1250927</int>
```

2.16 SENDWAPPUSHESINGLE

Send a single WAP message using a session token.

2.16.1 SOAP

Code Snippet:

```
int MessageID = _service. SendWAPPushSingle (sessionToken, recipient,message);
```

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- recipient (string) : A valid recipient number.
- message (string) : The WAP message to send to the recipient.

Returns: The message ID of the newly created message (value).

2.16.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/SendWAPPushSingle?session_token=123456789&mobile=27721234567&message=cellfindportal.co.za

Parameters:

- sessionToken (string) : A valid logged in sessionToken as generated by the Gateway web service
- recipient (string) : A valid recipient number.
- message (string) : The WAP message to send to the recipient.

Returns: The message ID of the newly created message (value) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<int xmlns="http://tempuri.org/">1250928</int>
```

2.17 GETINCOMINGMESSAGESBYSHORTCODEONCE

Get a list of incoming SMS messages by short code and mark the message as being reported on. This will prevent the message from being pulled on the subsequent calls.

2.17.1 SOAP

Code Snippet:

```
dataSet = _service.GetIncomingMessagesByShortCodeOnce(sessionToken, shortCode);
```

Parameters:

- shortCode : A valid short code number to get the received messages that have not been reported on.
- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid).

2.17.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessagesByShortCodeOnce?session_token=123456789&short_code=31100

Parameters:

- shortCode : A valid short code number to get the received messages that have not been reported on.
- sessionToken : A valid logged in sessionToken as generated by the Gateway web service.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid).

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://tempuri.org/">
```

```

<xs:schema id="ReturnDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="ReturnDataSet" msdata:isDataSet="true" msdata:UseCurrentLocale="true">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="ReplyDataTable">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Reply_Message_ID" type="xs:string" minOccurs="0" />
              <xs:element name="Destination_No" type="xs:string" minOccurs="0" />
              <xs:element name="Message_ID" type="xs:string" minOccurs="0" />
              <xs:element name="Message" type="xs:string" minOccurs="0" />
              <xs:element name="Originator" type="xs:string" minOccurs="0" />
              <xs:element name="Protocol_ID" type="xs:string" minOccurs="0" />
              <xs:element name="Receive_Time" type="xs:string" minOccurs="0" />
              <xs:element name="Reported" type="xs:string" minOccurs="0" />
              <xs:element name="Report_Date" type="xs:string" minOccurs="0" />
              <xs:element name="Network" type="xs:string" minOccurs="0" />
              <xs:element name="Tag" type="xs:string" minOccurs="0" />
              <xs:element name="User_ID" type="xs:string" minOccurs="0" />
              <xs:element name="smsid" type="xs:string" minOccurs="0" />
              <xs:element name="Push_Date" type="xs:string" minOccurs="0" />
              <xs:element name="Pushed" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-
microsoft-com:xml-diffgram-v1">
  <ReturnDataSet xmlns="">
    <ReplyDataTable diffgr:id="ReplyDataTable1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
      <Reply_Message_ID>893</Reply_Message_ID>
      <Destination_No>31180</Destination_No>
      <Message_ID>0</Message_ID>
      <Message>Stop</Message>
      <Originator>0822929390</Originator>
      <Protocol_ID>1</Protocol_ID>
      <Receive_Time>8/11/2008 10:00:51 PM</Receive_Time>
      <Reported>0</Reported>
      <Network>1</Network>
      <Tag />
      <User_ID>1</User_ID>
      <smsid>9290316</smsid>
    </ReplyDataTable>
  </ReturnDataSet>
</diffgr:diffgram>
</DataSet>

```

2.18 GETINCOMINGMESSAGESBYSHORTCODEONCEADMIN

Get a list of incoming SMS messages by short code and mark the message as being reported on. This call is only available to the admin user of the account. Any of the short codes under the account can be sent in for a list of received messages. This will prevent the message from being pulled on the subsequent calls.

2.18.1 SOAP

Code Snippet:

```
dataSet = _service.GetIncomingMessagesByShortCodeOnceAdmin(sessionToken, shortCode);
```

Parameters:

- **shortCode** : A valid short code number to get the received messages that have not been reported on. This short code has been assigned to the account.
- **sessionToken** : A valid logged in sessionToken as generated by the Gateway web service. This has to be the admin user for the account.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid).

2.18.2 HTTP GET

Code Snippet:

https://www.cellportal.co.za/gatewaywebservice/Service.asmx/GetIncomingMessagesByShortCodeOnceAdmin?session_token=123456789&short_code=33110

Parameters:

- **shortCode** : A valid short code number to get the received messages that have not been reported on. This short code has been assigned to the account.
- **sessionToken** : A valid logged in sessionToken as generated by the Gateway web service. This has to be the admin user for the account.

Returns : A dataset with the incoming message detail (Reply_Message_ID, Destination_No, Message_ID, Message, Originator, Protocol_ID, Receive_Time, Reported, Report_Date, Network, Tag, User_ID, smsid) in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<DataSet xmlns="http://tempuri.org/">
  <xs:schema id="ReturnDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="ReturnDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="ReplyDataTable">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Reply_Message_ID" type="xs:string" />
                <xs:element name="Destination_No" type="xs:string" minOccurs="0" />
                <xs:element name="Message_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Message" type="xs:string" minOccurs="0" />
                <xs:element name="Originator" type="xs:string" minOccurs="0" />
                <xs:element name="Protocol_ID" type="xs:string" minOccurs="0" />
                <xs:element name="Receive_Time" type="xs:string" minOccurs="0" />
                <xs:element name="Reported" type="xs:string" minOccurs="0" />
                <xs:element name="Report_Date" type="xs:string" minOccurs="0" />
                <xs:element name="Network" type="xs:string" minOccurs="0" />
                <xs:element name="Tag" type="xs:string" minOccurs="0" />
                <xs:element name="User_ID" type="xs:string" minOccurs="0" />
                <xs:element name="smsid" type="xs:string" minOccurs="0" />
                <xs:element name="Push_Date" type="xs:string" minOccurs="0" />
                <xs:element name="Pushed" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

```
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-
microsoft-com:xml-diffgram-v1">
  <ReturnDataSet xmlns="">
    <ReplyDataTable diffgr:id="ReplyDataTable1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
      <Reply_Message_ID>106396</Reply_Message_ID>
      <Destination_No>35610</Destination_No>
      <Message_ID>0</Message_ID>
      <Message>Test message 1</Message>
      <Originator>27741234567</Originator>
      <Protocol_ID>1</Protocol_ID>
      <Receive_Time>8/17/2009 4:49:40 AM</Receive_Time>
      <Reported>0</Reported>
      <Network>3</Network>
      <Tag>0</Tag>
      <User_ID>1</User_ID>
      <smsid>1234567</smsid>
    </ReplyDataTable>
    <ReplyDataTable diffgr:id="ReplyDataTable2" msdata:rowOrder="1" diffgr:hasChanges="inserted">
      <Reply_Message_ID>103788</Reply_Message_ID>
      <Destination_No>35610</Destination_No>
      <Message_ID>0</Message_ID>
      <Message>Test message 2</Message>
      <Originator>27821234567</Originator>
      <Protocol_ID>1</Protocol_ID>
      <Receive_Time>8/13/2009 11:21:40 PM</Receive_Time>
      <Reported>0</Reported>
      <Network>1</Network>
      <Tag>0</Tag>
      <User_ID>1</User_ID>
      <smsid>1234567</smsid>
    </ReplyDataTable>
  </ReturnDataSet>
</diffgr:diffgram>
</DataSet>
```

2.19 SENDOTASETTINGS

Send a single or multiple OTA settings to recipient MSISDNs. The method call on the Cellfind web service will expect a valid session token received from the Cellfind system when logging in using the Login method call detailed in section 2.1. The OTAStruct is a publicly available structure on the Cellfind web service that needs to be submitted in array format with all the mandatory fields completed. Default values will be used for the non-mandatory fields.

2.19.1 SOAP

Code Snippet (C# example):

```
//Create the web service object
OTAService.Service _service = new OTAService.Service();

//Create an array for the objects to be passed to the service
OTAService.OTAStruct[] settingsList = new OTAService.OTAStruct[OTAList.Count];

for (int i = 0; i < OTAList.Count; i++)
{
    OTASettings setting = (OTASettings)OTAList[i];

    //Create the objects
    OTAService.OTAStruct OTASetting = new OTAService.OTAStruct();
    OTASetting.BEARER = OTAService.BEARER.GPRS;
    OTASetting.NAME = setting.Name;
    OTASetting.APN = setting.APN;
    OTASetting.PPP_AUTHNAME = setting.Username;
    OTASetting.PPP_AUTHSECRET = setting.Password;
    OTASetting.PPP_AUTHTYPE = OTAService.PPP_AUTHTYPE.PAP;

    //Add the object to the array
    settingsList[i] = OTASetting;
}

//Submit the array to the web service
settingsList = _service.SendOTASettings("sessionToken", settingsList);
```

Parameters:

- sessionToken : A valid logged in sessionToken as generated by the Gateway web service. This has to be the admin user for the account.
- settings : An array of OTA settings (OTAStruct[] objects) to be submitted for delivery to a recipient MSISDNs.

Returns : An OTA structure array with the status of the submitted settings and the original data submitted to the service. Please note that the "STATUS" would be the status of submission of the settings to the system/network. If initial submission to the system was successful, the OTAStruct.STATUS will contain a value of "OK" (STATUS enumerator) and "FAILED" on submission failure for a specific setting.

OTA Structure:

Parameter	Type	Defaults
NAME*	string	
APN*	string	
BEARER	BEARER (enumerator)	GPRS
PPP_AUTHNAME*	string	
PPP_AUTHSECRET*	string	
PPP_AUTHTYPE	PPP_AUTHTYPE (enumerator)	PAP

PPP_LOGINTYPE	PPP_LOGINTYPE (enumerator)	AUTOMATIC
PROXY	string	
PORT	PORT	9200 (CONNECTIONLESS)
PROXY_LOGINTYPE	PROXY_LOGINTYPE	AUTOMATIC
PROXY_AUTHNAME	string	
PROXY_AUTHSECRET	string	
CSD_CALLTYPE	CSD_CALLTYPE	ANALOGUE
CSD_CALLSPEED	CSD_CALLSPEED	AUTO
CSD_DIALSTRING	string	
ISP_NAME	string	
STATUS	string	OK

*Mandatory field on the OTA structure

OTA Structure XML schema:

```
<s:complexType name="OTAstruct">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="NAME" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="APN" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="BEARER" type="tns:BEARER" />
    <s:element minOccurs="0" maxOccurs="1" name="PPP_AUTHNAME" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="PPP_AUTHSECRET" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="PPP_AUTHTYPE" type="tns:PPP_AUTHTYPE" />
    <s:element minOccurs="1" maxOccurs="1" name="PPP_LOGINTYPE" type="tns:PPP_LOGINTYPE" />
    <s:element minOccurs="0" maxOccurs="1" name="PROXY" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="PORT" type="tns:PORT" />
    <s:element minOccurs="1" maxOccurs="1" name="PROXY_LOGINTYPE" type="tns:PROXY_LOGINTYPE" />
    <s:element minOccurs="0" maxOccurs="1" name="PROXY_AUTHNAME" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="PROXY_AUTHSECRET" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="CSD_CALLTYPE" type="tns:CSD_CALLTYPE" />
    <s:element minOccurs="1" maxOccurs="1" name="CSD_CALLSPEED" type="tns:CSD_CALLSPEED" />
    <s:element minOccurs="0" maxOccurs="1" name="CSD_DIALSTRING" type="s:string" />
    <s:element minOccurs="0" maxOccurs="1" name="ISP_NAME" type="s:string" />
    <s:element minOccurs="1" maxOccurs="1" name="STATUS" type="tns:STATUS" />
  </s:sequence>
</s:complexType>
```

2.20 SENDMMSSINGLEUPLOADFILES

Send a single MMS using a session token. All MMS details and slides are uploaded in data tables containing byte arrays for slide media.

Code Snippet:

```
DataTable SlidesDataTable = new DataTable();
DataTable MMSDetailDataTable = new DataTable();
DataColumn myDataColumn;
DataRow row;

//Recipient MSISDNs comma delimited
String RecipientList = "27721234567, 27821234567";

//SessionToken
String SessionToken = "YourValidSessionToken";

//Define the MMS Detail DataTable
MMSDetailDataTable.TableName = "MMSDetailDataTable";

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.String");
myDataColumn.ColumnName = "Subject";
```

```
MMSDetailDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.Boolean");
myDataColumn.ColumnName = "Scheduled";
MMSDetailDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.String");
myDataColumn.ColumnName = "SendDate";
MMSDetailDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.String");
myDataColumn.ColumnName = "SendTime";
MMSDetailDataTable.Columns.Add(myDataColumn);

//Define the Slides Detail DataTable
SlidesDataTable.TableName = "SlidesDataTable";

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.String");
myDataColumn.ColumnName = "Text";
SlidesDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.String");
myDataColumn.ColumnName = "SlideDuration";
SlidesDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.Double");
myDataColumn.ColumnName = "SlideSize";
SlidesDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.Byte[]");
myDataColumn.ColumnName = "SlideImageBytes";
SlidesDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.Byte[]");
myDataColumn.ColumnName = "SlideAudioBytes";
SlidesDataTable.Columns.Add(myDataColumn);

myDataColumn = new DataColumn();
myDataColumn.DataType = Type.GetType("System.Byte[]");
myDataColumn.ColumnName = "SlideVideoBytes";
SlidesDataTable.Columns.Add(myDataColumn);

//Populate the MMS Detail DataTable
MMSDetailDataTable.Clear();
row = MMSDetailDataTable.NewRow();

row["Subject"] = "Happy New Year";
row["Scheduled"] = true;
row["SendDate"] = "2013-01-01" ;
row["SendTime"] = "00:00";

//Only one row submitted in the Detail DataTable
MMSDetailDataTable.Rows.Add(row);

//Populate the MMS Slide DataTable
row = SlidesDataTable.NewRow();

//Slide Text
```

```
row["Text"] = "Wishing you a happy new year!";

//Slide Duration
String slideDuration = 10;
row["SlideDuration"] = slideDuration + " second(s)";

//Slide Image Bytes
byte[] imageBytes = System.IO.File.ReadAllBytes(imageFileName);
row["SlideImageBytes"] = imageBytes;

//Slide Audio Bytes
byte[] audioBytes = System.IO.File.ReadAllBytes(audioFileName);
row["SlideAudioBytes"] = audioBytes;

//Slide Video Bytes
byte[] videoBytes = System.IO.File.ReadAllBytes(videoFileName);
row["SlideVideoBytes"] = videoBytes;

//Multiple slides can be added to the SlideDataTable in the same way
SlidesDataTable.Rows.Add(row);

GatewayWS.Service _WSservice = new GatewayWS.Service();
String response = _WSservice.SendMMSSingleUploadFiles(SessionToken
                                                    MMSDetailDataTable,
                                                    SlidesDataTable,
                                                    RecipientsList);
```

Parameters:

- SessionToken : A valid logged in sessionToken as generated by the Gateway web service
- MMSDetailDataTable : A Datatable containing the MMS detail.
- SlideDataTable : A Datatable containing the MMS slide detail.
- RecipientsList: A Comma delimited string of MSISDNs

Returns : A String with the status of the MMS upload.

2.21 SENDSINGLEHLRLOOKUPREQUEST

Send a single MMS using a session token. All MMS details and slides are uploaded in data tables containing byte arrays for slide media.

Code Snippet:

```
//SessionToken
String SessionToken = "YourValidSessionToken";
String msisdn = "072xxxxxx";

GatewayWS.Service _WSservice = new GatewayWS.Service();
String response = _WSservice.SendSingleHLRLookupRequest(SessionToken,msisdn);
```

Parameters:

- SessionToken : A valid logged in sessionToken as generated by the Gateway web service
- msisdn : The mobile number you would like to do a HLR Lookup on

Returns : A String with the Result of the HLR Lookup

Example Response:

Definition: MSISDN;HLR_ID;STATUS;ERROR_CODE


```
<string>
2772xxxxxxx;00000000000-11111-2222;DELIVRD;0;
</string>
```

Statuses may be:

“DELIVRD” in case HLR is executed fine,

“UNDELIV” in case of error (will return status code),

“UNKNOWN” in case of any other system related error

Error Codes Table:

Error_Code	Description	IsPermanent
1	EC_UNKNOWN_SUBSCRIBER	Yes
5	EC_UNIDENTIFIED_SUBSCRIBER	No
6	EC_ABSENT_SUBSCRIBER_SM	No
9	EC_ILLEGAL_SUBSCRIBER	Yes
11	EC_TELESERVICE_NOT_PROVISIONED	Yes
12	EC_ILLEGAL_EQUIPMENT	Yes
13	EC_CALL_BARRED	No
21	EC_FACILITY_NOT_SUPPORTED	No
27	EC_ABSENT_SUBSCRIBER	No
31	EC_SUBSCRIBER_BUSY_FOR_MT_SMS	No
32	EC_SM_DELIVERY_FAILURE	No
33	EC_MESSAGE_WAITING_LIST_FULL	No
34	EC_SYSTEM_FAILURE	No
35	EC_DATA_MISSING	Yes
36	EC_UNEXPECTED_DATA_VALUE	Yes
256	EC_SM_DF_memoryCapacityExceeded	No
257	EC_SM_DF_equipmentProtocolError	No
258	EC_SM_DF_equipmentNotSM_Equipped	No
259	EC_SM_DF_unknownServiceCentre	No
260	EC_SM_DF_sc_Congestion	No
261	EC_SM_DF_invalidSME_Address	No
262	EC_SM_DF_subscriberNotSC_Subscriber	No
500	EC_PROVIDER_GENERAL_ERROR	No
502	EC_NO_RESPONSE	No
503	EC_SERVICE_COMPLETION_FAILURE	No
504	EC_UNEXPECTED_RESPONSE_FROM_PEER	No
507	EC_MISTYPED_PARAMETER	No
508	EC_NOT_SUPPORTED_SERVICE	No
509	EC_DUPLICATED_INVOKE_ID	No
511	EC_INITIATING_RELEASE	No
1024	EC_OR_appContextNotSupported	No
1025	EC_OR_invalidDestinationReference	No
1026	EC_OR_invalidOriginatingReference	No

1027	EC_OR_encapsulatedAC_NotSupported	No
1028	EC_OR_transportProtectionNotAdequate	No
1030	EC_OR_potentialVersionIncompatibility	No
1031	EC_OR_remoteNodeNotReachable	No
1152	EC_NNR_noTranslationForAnAddressOfSuchNatur	No
1153	EC_NNR_noTranslationForThisSpecificAddress	No
1154	EC_NNR_subsystemCongestion	No
1155	EC_NNR_subsystemFailure	No
1156	EC_NNR_unequippedUser	No
1157	EC_NNR_MTPfailure	No
1158	EC_NNR_networkCongestion	No
1159	EC_NNR_unqualified	No
1160	EC_NNR_errorInMessageTransportXUDT	No
1161	EC_NNR_errorInLocalProcessingXUDT	No
1162	EC_NNR_destinationCannotPerformReassemblyXUDT	No
1163	EC_NNR_SCCPfailure	No
1164	EC_NNR_hopCounterViolation	No
1165	EC_NNR_segmentationNotSupported	No
1166	EC_NNR_segmentationFailure	No
1281	EC_UA_userSpecificReason	No
1282	EC_UA_userResourceLimitation	No
1283	EC_UA_resourceUnavailable	No
1284	EC_UA_applicationProcedureCancellation	No
1536	EC_PA_providerMalfunction	No
1537	EC_PA_supportingDialogOrTransactionReleased	No
1538	EC_PA_ressourceLimitation	No
1539	EC_PA_maintenanceActivity	No
1540	EC_PA_versionIncompatibility	No
1541	EC_PA_abnormalMapDialog	No
1793	EC_NC_responseRejectedByPeer	No
1794	EC_NC_abnormalEventReceivedFromPeer	No
1795	EC_NC_messageCannotBeDeliveredToPeer	No
1796	EC_NC_providerOutOfInvoke	No
2048	EC_TIME_OUT	No
2049	EC_IMSI_BLACKLISTED	Yes
2050	EC_DEST_ADDRESS_BLACKLISTED	Yes
2051	EC_InvalidMscAddress	No
4096	EC_InvalidPduFormat	Yes
4100	EC_Cancelled	Yes
4101	EC_ValidityExpired	Yes

3 PUSHSERVICESPECIFICATIONS

With the pushing service the following methods are used to achieve the desired results. The following methods can be implemented in the **Service** namespace on a client's public web service to get data pushed to the client's system as it changes. All the methods do not have to be implemented, but the ones that will be implemented must be registered with Cellfind. Making use of the pushing method means that any changes to data will be pushed to clients as it changes, eliminating the need to poll the Cellfind system for data and alleviating network/internet traffic. This also prevents polling limits being implemented.

Public Method Call	Note
IncomingMessagePushParamAll	Get incoming messages (MO) on premium rated/short code numbers
IncomingReplyMessagePushParamAll	Get incoming reply messages (MO) that were sent back on an original outgoing message
DeliveryReceiptPushParam	Get delivery statuses on outgoing messages (MT)

C# Code Snippets and more detail:

IncomingMessagePushParamAll

This web service call is used to receive messages that were received on a premium rated number – there is no tag as a premium rated number will never come in with a tag

```
public bool IncomingMessagePushParamAll(int Reply_Message_ID, string Destination_No, string Message,
string Originator, DateTime Receive_Time, string Network)
{
    *Do what needs to be done with the data
    *Return True/False
}
```

Parameters:

- **Reply_Message_ID:** The unique incoming Message ID on the Cellfind system.
- **Destination_No:** Number that the message was sent to.
- **Message:** The message text received from the sender.
- **Originator:** MSISDN number for the sender.
- **Receive_Time:** The date and time that the message was received.
- **Network:** The network that the message was sent/received on.

Return : Return **True/False**. Returning **False** to the Cellfind Messaging System will leave the messages in the queue to be resubmitted to the client system. Returning **True** will mark the message as being submitted to the client system successfully and will not be queued for resubmission.

IncomingReplyMessagePushParamAll

This web service call is used to receive reply messages on outgoing messages – This will include the tag of the outgoing message as well as the unique Message_ID for the original outgoing message.

```
public bool IncomingReplyMessagePushParamAll(int Reply_Message_ID, string Destination_No, string
Message, string Originator, DateTime Receive_Time, int Message_ID, string Network, string Tag)
{
    *Do what needs to be done with the data
    *Return True/False
}
```

Parameters:

- **Reply_Message_ID**: The unique incoming Message ID on the Cellfind system.
- **Destination_No**: Number that the message was sent to.
- **Message**: The message text received from the sender.
- **Originator**: MSISDN number for the sender.
- **Receive_Time**: The date and time that the message was received.
- **Message_ID**: The original unique Messages ID of the outgoing message.
- **Network**: The network that the message was sent/received on.
- **Tag**: The tag of the outgoing/incoming message.

Return : Return **True/False**. Returning **False** to the Cellfind Messaging System will leave the messages in the queue to be resubmitted to the client system. Returning **True** will mark the message as being submitted to the client system successfully and will not be queued for resubmission.

DeliveryReceiptPushParam

This web service call is used to push the delivery statuses of sent messages when/if they are delivered.

```
public bool DeliveryReceiptPushParam(int Message_ID, DateTime Delivery_Time)
{
    *Do what needs to be done with the data
    *Return True/False
}
```

Parameters:

- **Message_ID**: The original unique Messages ID of the outgoing message.
- **Delivery_Time**: : The date and time that the message was delivered.

Return : Return **True/False**. Returning **False** to the Cellfind Messaging System will leave the delivery status in the queue to be resubmitted to the client system. Returning **True** will mark the delivery status as being submitted to the client system successfully and will not be queued for resubmission.

4 Possible Error Codes returned from the Web Service

Error	Error Code	Description
CallLimitReached	-121	
Duplicate	-120	
NoText	-119	
NotActive	-115	
InvalidMobileNumber	-112	
NotValidToken	-99	Not a valid token.
InsertError	-88	Failed to insert the data.
SelectError	-55	Failed to return valid data.
UserDoesNotExist	-33	The user does not exist.
UpdateError	-22	Failed to update the detail.
NotCellFAdmin	-11	Not a CellFind administrator.
NotAccountAdmin	-7	Not a valid account administrator.
ExclusionExists	-4	The added exclusion exists.
Success	-1	Successfully updated/inserted data.

5 Available number formats

Format	Note
82xxxxxxx	On the bulk CSV files only. Won't work on the web site or the direct web service calls.
082xxxxxxx	Standard number national format.
082 xxx xxxx	Numbers can have spaces in.
2782xxxxxxx	Compulsory format for international bulk CSV files
2782 xxx xxxx	Numbers can have spaces in.
+2782xxxxxxx	Compulsory format for international bulk CSV files
+2782 xxx xxxx	Numbers can have spaces in.
+4482xxxxxxx	Will not be sent out, but will be accepted by the system and marked as an invalid number.
+4482 xxx xxxx	Will not be sent out, but will be accepted by the system and marked as an invalid number

6 Available Gateway Web Service Functions

- [BulkSMSFileUpload](#)
Upload a CSV file for bulk sms messages
- [BulkSMSFileUploadScheduled](#)
Upload a CSV file for bulk sms messages that get scheduled
- [GetIncomingMessageDetailByID](#)
Get the detail for a specific incoming message by ID.
- [GetIncomingMessageIDsByUserDataSet](#)
Get a list of incoming SMS message IDs for a user - returns a dataset
- [GetIncomingMessagesByShortCode](#)
Get a list of incoming SMS messages by short code
- [GetIncomingMessagesByShortCodeOnce](#)
Get a list of incoming SMS messages by short code and mark the message as being reported on
- [GetIncomingMessagesByShortCodeOnceAdmin](#)
Get a list of incoming SMS messages by short code and admin user and mark the message as being reported on
- [GetIncomingMessagesByUser](#)
Get a list of incoming SMS messages by user
- [GetIncomingMessagesByUserDataSet](#)
Get a list of incoming SMS messages by user - returns a dataset
- [GetIncomingMessagesForDateRange](#)
Get a list of incoming SMS messages by User for a date range - returns a dataset
- [GetSentSMSMessageDetailByID](#)
Get the detail for sent SMS messages by Message ID- returns a dataset
- [GetSessionStatus](#)
Get the session status for a specific session.
- [LoadCustomerDetail](#)
Load the Customer detail by ID.
- [LoadUserProfile](#)
Load the logged in user's profile.
- [LoadUserProfileRights](#)
Load the logged in user's rights.
- [LoadUserProfileServices](#)
Load the logged in user's account services.
- [Login](#)
Log the user into the system and get a session token back.
- [LogOut](#)
Log the user out of the system and terminate the current session.