

Message Sending and Account Management

Version 4.5

Dated 04/06/2011

Intended Audience

Boom-SMS Users

Contents

Message Statements and Account Management	1
Intended Audience	1
API Location	1
Brief	1
Structure	2
Custom Data Types	2
Interface: Sending	2
Interface: Checking	3
Interface: Details	4
Example Code	7

API Location

The URL is: "<http://services.boom-sms.co.uk/accountservices/boomuser.svc>"

Brief

This document explains how a Boom-SMS user can integrate account checking and message sending within their application. This is a WCF API that uses a SOAP transport and can be implemented in any language that supports Soap.

This API exposes some custom data types in addition to functions that are designed to be used in combination. The details of both are in the next section. At the end of this document we have provided some example code in the Visual Basic .net language to aid you in integrating this with your software.

For customers who wish to take advantage of the Boom-SMS API without using the Microsoft .Net architecture we provide a number of ASP wrapper pages that encapsulate its functionality. These ASP pages accept posted data in the form of a query string. The root address for ASP wrappers for the Boom-User web-service is <http://services.boom-sms.co.uk/BoomUserASP/> this address should be followed by the function name and a query string that contains the values you wish to pass into the function. For all API calls documented below for which an ASP alternative exists an example query string is provided and **highlighted in red**.

We also recommend that users seeking to evaluate our API before integrating with their own software do so with the excellent free testing tool soapUI, made by eviware and available from soapui.org.

Structure

Custom Data Types

Enumerated type: Account_Operation_Result

Values

- 0. Success:** Indicates a successful account creation
- 1. InvalidPartner:** Invalid partner details
- 2. InvalidAuthentication:** Partner is not authorised to set-up accounts remotely
- 3. InvalidField:** Field name was not recognised
- 4. InvalidDateField:** A supplied Date value is not in a valid date format
- 5. FieldNotUnique:** A supplied value already exists in the database (e.g. **AccountID**)
- 6. EmptyValue:** Indicates an essential field was not supplied with a value or was supplied with an empty value.
- 7. InvalidEmail:** Indicates a supplied email address was not in a proper email format
- 8. InvalidCharacters:** Indicates an invalid was supplied in a field (e.g. | , # %)
- 9. InternalError:** Indicates an internal server error, please re-try and notify support if problem re-occurs.
- 10. PasswordMismatch:** Indicates that when attempting to change an account password the supplied existing password did not match the existing password of the account.

Enumerated type: CredCheck

Values

- 0. noMatch:** Indicates the specified username was not found.
- 1. partMatch:** Indicates the username was found but the supplied password was incorrect.
- 2. fullMatch:** Indicates the username was found and the supplied password was correct.

Class: searchcriteria

Note: all fields are optional, specify only desired values

Values

- mobto:** a string containing the destination mobile number
- msg:** a string containing the contents of the message that was sent
- dstatus:** value of enum delivery_status, possible values are 1 (pending), 2 (failed), 3 (succeeded), 4 (blank), 5(unknown) and 100(all).

Class: Smsclass

Values

- strAccount_ID:** A string containing your account ID.
- strPassword:** A string containing your account password.
- strUsername:** A string containing your account username.
- strMobTo:** An array of strings containing the destination mobile numbers.
- strMessage:** A string containing the body of the message.
- bNotify:** A Boolean variable containing true or false to indicate notification status (optional).
- strSenderEmail:** A string containing the e-mail address of the sender.

The Methods of this API are split into several interfaces, one encapsulating Sending functionality, another encapsulating Checking functionality and another encapsulating retrieving message and account information.

Interface: Sending

Function: XML_String_Send

Parameters

post_string: a string that contains the XML data to be sent.

Return

bFound: a Boolean value that indicates if a valid <sms.message> tag was found.

Function: SMS_Send_Object

Parameters

XMLobj: an array of objects of the smsclass class.

Return

bResult: a Boolean value that indicates success or failure. This return value indicates only the receipt of your message by our server; this is **not** a final delivery status.

ASP Alternative:

[Smsobject_Send/?account.id=testuser&password=testpass&mobile.to=07700900000&message=testing123¬ify=true&sender.email=someone@somewhere.com](#)

Function: XML_Send_String_Delayed

Parameters

post_string: a string that contains the XML data to be sent

TimeToSend: a DateTime variable that indicates the time that the message should be sent (in GMT).

Return

bFound: a Boolean value that indicates if a valid <sms.message> tag was found.

Function: SMS_Send_Object_Delayed

Parameters

XMLobj: an array of objects of the smsclass class.

TimeToSend: a DateTime variable that indicates the time that the message should be sent (in GMT).

Return

bResult: a Boolean value that indicates success or failure. This return value indicates only the receipt of your message by our server; this is **not** a final delivery status.

ASP Alternative:

[Smsobject_Send_Delayed/?account.id=testuser&password=testpass&mobile.to=07700900000&message=testing123&delay=29.05.1988¬ify=true&sender.email=someone@somewhere.com](#)

Interface: Checking

Function: Credential_Check

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

Return

bResult: A value of the enumerated type CredCheck (see above for specifics).

ASP Alternative:

[Credential_Check/?Username=testuser&Password=testpass](#)

Function: Credit_Check

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

Return

bResult: An integer value containing the number of credits on the account.

ASP Alternative:

[Credit_Check/?Username=testuser&Password=testpass](#)

Interface: Details

Note: The Get_Messages functions can timeout if being used to query a large amount of data. We therefore recommend breaking large queries down into smaller chunks by date.

Function: Get_Account_Details_Dataset

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

Return

RetData: A system.data.dataset variable that contains all stored information regarding your account.



Simple Solutions Making a Big Difference

Function: Get_Account_Details_XML

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

Return

RetData: A string value containing all stored information regarding your account formatted in XML.

ASP Alternative:

[Account_Details_XML/?Username=testuser&Password=testpass](#)

Function: Get_Messages_Dataset

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

SearchCriteria: an object of the searchcriteria class described above.

Sdate: a DateTime variable, used to indicate oldest messages to search.

Edate: a DateTime variable, used to indicate newest messages to search.

Return

RetData: A system.data.dataset variable that contains your search results

Function: Get_Messages_XML

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

SearchCriteria: an object of the searchcriteria class described above.

Sdate: a DateTime variable, used to indicate oldest messages to search.

Edate: a DateTime variable, used to indicate newest messages to search.

Return

RetData: A string value that contains your search results formatted in XML.

ASP Alternative:

[User_Messages_XML/?Username=testuser&Password=testpass](#)

Function: Get_Contacts_Dataset

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

Return

RetData: A system.data.dataset variable that contains all stored information regarding your contacts.

Function: Get_Contacts_XML

Parameters

UserName: A string containing your account name.

Passwd: A string containing your account password.

Return

RetData: A string value containing all stored information regarding your contacts formatted in XML.

ASP Alternative:

[User_Contacts_XML/?Username=testuser&Password=testpass](#)

Function: Add_Contact

Parameters

Username: A string containing your account name.

Password: A string containing your account password.

FirstName: A string containing the desired contacts first name.

Email: A string containing the desired contact email address.

Tel: A string containing the desired contact telephone number.

LastName: Optional. A string containing the desired contacts last name.

Mob: Optional. A string containing the desired contact mobile number.

Address: Optional. A string containing the desired contact address.

Postcode: Optional. A string containing the desired contact postal code.

NotifySent: Optional. A Boolean value indicating if this contact should receive sent message notifications.

NotifyFail: Optional. A Boolean value indicating if this contact should receive failed message notifications.

NotifySent: Optional. A Boolean value indicating if this contact should receive sent message notifications.

NotifyInvoice: Optional. A Boolean value indicating if this contact should receive Invoices.

NotifyWP: Optional. A Boolean value indicating if this contact should receive World Pay transaction notifications.

NotifyWeekly: Optional. A Boolean value indicating if this contact should receive weekly statements.

NotifyLowcreds: Optional. A Boolean value indicating if this contact should receive low credits notifications.

Return

b: A Boolean value indicating success or failure in adding this contact.

ASP Alternative:

[Add_Contact/?Username=testuser&Password=testpass&FirstName=fname&Email=someone@somewhere.com&Tel=01189998819991197253&LastName=lname&Mob=07000900000&Address=42aroad&Postcode=GU164QA&NotifSent=false&NotifFail=true&NotifReceived=false&NotifInvoice=true&NotifWorldpay=false&NotifWeekly=true&NotifCreds=false](#)

Function: Remove_Contact

Parameters

Username: A string containing your account name.

Password: A string containing your account password.

ContactName: A string containing the desired contacts first name.

Return

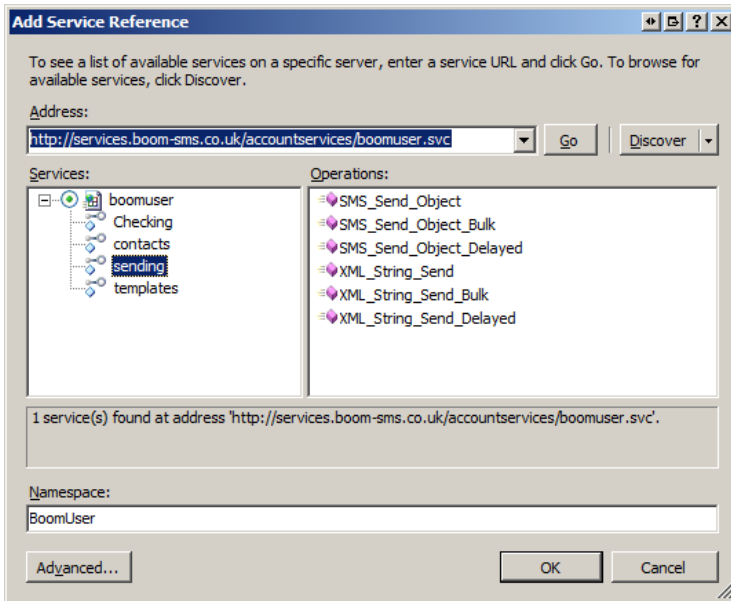
Contact_Create_Result: a value of the enumerated type Account_Operation_result (see above for specifics).

ASP Alternative:

[Remove_Contact/?Username=testuser&Password=testpass&ContactName=person](#)

Example Code

The below example code is written in Visual Basic .net and is confirmed to work with the Boom-SMS API as of the date of this document. All clients must first register the boom-sms service reference in their project as below.





Simple Solutions Making a Big Difference

```
Sub Main()

    'Create the objects that encapsulate the functionality of the API'
    Console.WriteLine("Creating service references...")
    Dim BoomUser_Sending As New BoomUser.SendingClient("SendingWCF") 'BoomUser.SendingClient provides access to message
    sending functions
    Dim BoomUser_Checking As New BoomUser.CheckingClient("CheckingWCF") 'BoomPartner.StatementsClient provides access
    to statement functions

    'Create your messages'
    Console.WriteLine("Creating message...")
    Dim XMLstring As String =
    "<messages><sms.message><account.id>boomtest</account.id><password>testpw</password><username>testun</username><mobile.to>0
    123456789</mobile.to><mobile.to>0987654321</mobile.to><message>another test
    message</message><notify>1</notify><sender.email>test@boom-sms.co.uk</sender.email></sms.message></messages>"
    Dim messageobjects(0 To 1) As BoomUser.smsclass
    messageobjects(0) = createobj()
    messageobjects(1) = createobj()
    Dim timetosend As DateTime = "31/03/2010"

    'Run functions of the Sending interface'
    Console.WriteLine("Running User Sending functions...")
    'send messages using an XML string
    BoomUser_Sending.XML_String_Send(XMLstring)
    Console.WriteLine("XML_String_Send OK.")
    'send a delayed message using an XML string
```



Simple Solutions Making a Big Difference

```
BoomUser_Sending.XML_String_Send_Delayed(XMLstring, timetosend)
Console.WriteLine("XML_String_Send_Delayed OK.")
'send messages using an smsclass array
BoomUser_Sending.SMS_Send_Object(messageobjects)
Console.WriteLine("SMS_Send_Object OK.")
'send a delayed message using an smsclass array
BoomUser_Sending.SMS_Send_Object_Delayed(messageobjects, timetosend)
Console.WriteLine("SMS_Send_Object_Delayed OK.")
Console.WriteLine()

'Run functions of the Checking interface'
'Run functions of the Checking interface'

Console.WriteLine("Running Partner Checking functions...")
'check credits of an account
BoomUser_Checking.Credit_Check("testUN", "testPW")
Console.WriteLine("Credit_Check OK.")
'check credentials of an account
Dim Check_Result As BoomUser.CredCheck = BoomUser_Checking.Credential_Check("testUN", "testPW")
Console.WriteLine("Credential_Check OK.")
Console.ReadLine()

End Sub
```

```
Function createobj() As BoomUser.smsclass

'specify message data
Dim smsmessage As New BoomUser.smsclass()
smsmessage.Account_ID = "boomtest"
smsmessage.Password = "testpw"
smsmessage.Username = "testun"
smsmessage.Mobile_To = getnumbers()
smsmessage.Message = "another test message"
smsmessage.Notify = True
```



Simple Solutions Making a Big Difference

```
    smsmessage.Sender_Email = "test@boom-sms.co.uk"  
    Return smsmessage
```

```
End Function
```

```
Private Function getnumbers() As String()
```

```
    'populate an array of telephone numbers
```

```
    Dim nstring(0 To 1) As String
```

```
    nstring(0) = "0123456789"
```

```
    nstring(1) = "0987654321"
```

```
    Return nstring
```

```
End Function
```