PowerText SOAP V2 API

Documentation
Rev 2.01

Author: David Wales

Last Updated: 01/12/2014

Table of Contents

1. Admin API

Admin API allows access to multiple SOAP login details. This is useful where an external source, such as the Project Manager, needs to make SOAP calls on data across multiple client accounts.

File:https://secure.powertext.co.uk/SOAP/V2/Admin.wsdl**Namespace:** AdminSOAP

Method: getSoapLogin

Your private "key" is matched against the requesting IP address, only nominated IPs have access to this function. The "key" is also matched against the "accountID" passed over in the request; access can be limited to specific accountID's.

1.1.1 Example: getSoapLogin

Function:`getSoapLogin(key As String, accountID As Integer) As AdminSOAP.loginDetails`

.NET

```
Dim myAdmin As New AdminSOAP.Admin
    Dim myDetails As New AdminSOAP.loginDetails
    myDetails = myAdmin.getSoapLogin("your private key", 3)
    MsgBox("user: " & myDetails.username & " pass:" & myDetails.password)
```

.PHP

```
    ini_set("soap.wsdl_cache_enabled", "0");
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Admin.wsdl";
   $client = new SoapClient($wsdl);
    $myDetails = $client->getSOAPLogin("your private key", 3);

    if($myDetails instanceof SoapFault) {
        die("SoapFault: ".$myDetails->getMessage());
    } else {
        echo "user: ".$myDetails->username." pass: ".$myDetails->password;
    }
```

RETURNS

```
object->username As String
object->password As String
```

Method: getSystemAuth

Authenticate a PowerText System username and password and return the PowerText System user data.

1.2.1 Example: getSystemAuth

Function:`getSystemAuth (username As String, password As String, pt_username As String, pt_password As String) As AdminSOAP.authDetails`

.NET

```
    Try
        Dim myService As New AdminSOAP.Admin
        Dim myUser As New AdminSOAP.authDetails
        myUser = myService.getSystemAuth("username","password",
                                         "pt_username", "pt_password")
        If String.IsNullOrEmpty(myUser.username) Then
            MsgBox("Failed to find a result.")
        Else
            MsgBox("Success, PowerText userID: " & myUser.userid)
        End If
Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
    ini_set("soap.wsdl_cache_enabled", "0");
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Admin.wsdl";
    $client = new SoapClient($wsdl);
    $myDetails = $client->getSystemAuth("username","password",
"user_username", "user_password");
    if($myDetails instanceof SoapFault) {
        die("SoapFault: ".$myDetails->getMessage());
    } else {
        echo "user: ".$myDetails->username." pass: ".$myDetails->password;
}
```

RETURNS

```
object->userid As Integer
object->roleid As Integer
object->unitid As Integer
object->username As String
object->password As String
object->firstname As String
object->lastname As String
object->mobilenumber As String
object->email As String
object->telephone As String
object->createdtimestamp As Integer
object->candelete As String (Y/N)
object->ptadminuser As String (Y/N)
```

Method: getInvoiceDetails

Returns a given PowerText invoiceID data set.

1.3.1 Example: getInvoiceDetails

Function:`getInvoiceDetails(key As String, invoiceID As Integer) As AdminSOAP.invoiceDetails`

.NET

```
Dim myAdmin As New AdminSOAP.Admin
    Dim myDetails As New AdminSOAP.invoiceDetails
    myDetails = myAdmin.getInvoiceDetails("key", 1234)
MsgBox("purchase ref: " & myDetails. purchaseref)
```

.PHP

```
    ini_set("soap.wsdl_cache_enabled", "0");
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Admin.wsdl";
    $client = new SoapClient($wsdl);
    $myDetails = $client->getInvoiceDetails("key", 1234);

    if($myDetails instanceof SoapFault) {
        die("SoapFault: ".$myDetails->getMessage());
    } else {
        echo "purchase ref: ".$myDetails->purchaseref;
    }
```

RETURNS

```
object->invoiceid As Integer
object->ptorderid As Integer
object->unitid As Integer
object->accountid As Integer
object->createdtime As Integer
object->purchaseref As String
object->paid As String
object->createdby As Integer
object->issent As String
object->senttosage As String
object->vat As Decimal
object->lines As Array of Object (o->invoicelineid As Integer, o->invoiceid As
Integer, o->itemid As Integer, o->description As String, o->qty As Integer,
o->quantityunitid As Integer, o->costperunit As Decimal, o->totalcost As
Decimal, o->financialtransactionid As Integer, o->createdtime As Integer)
```

Method: getSystemUsers

Returns all PowerText system users for logged in account.

1.4.1 Example: getSystemUsers

Function:`getSystemUsers (username As String, password As String) As AdminSOAP.authDetails()`

.NET

```
    Dim myService As New AdminSOAP.Admin
    Dim myUsers() As AdminSOAP.authDetails
    myUsers = myService.getSystemUsers("username", "password")
    If myUsers.Length >= 1 Then
        MsgBox("Found results.")
    Else
        MsgBox("No Results")
    End If
```

.PHP

```
    ini_set("soap.wsdl_cache_enabled", "0");
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Admin.wsdl";
    $client = new SoapClient($wsdl);
    $myUsers = $client->getSystemUsers("username", "password");
    echoprint_r($myUsers,1);
```

RETURNS

```
Array of Object (
  object->userid As Integer
  object->roleid As Integer
  object->unitid As Integer
  object->username As String
  object->password As String
  object->firstname As String
  object->lastname As String
  object->mobilenumber As String
  object->email As String
  object->telephone As String
  object->createdtimestamp As Integer
  object->candelete As String (Y/N)
  object->ptadminuser As String (Y/N)
)
```

2. Contact API

The Contact API provides methods for accessing contact data in both read and write. There are also methods for searching contact records using different criteria.

File:https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl**Namespace:** ContactSOAP

2.1 Method: createContact

This function can be used to create new contacts, if a contact exists with a matching email address, mobile number or combination of address fields and dob field the contact data will be merged in to the existing contact record. You should use the updateContact method if you have a contactID available and wish to update a specific contact record.

2.1.1 Example: createContact

Function:`createContact (username As String, password As String, contactData As ContactSOAP.contactObject, groups() As Integer, udfs() As ContactSOAP.UDF) As Integer`

.NET

```
Dim contactObj As New ContactSOAP.contactObject
        contactObj.sourceid = "7910"
        contactObj.firstname = "Joe"
        contactObj.lastname = "Blogs"
        contactObj.email = "me@example.com"
        contactObj.mobilenumber = "447911111111"
        contactObj.allowsms = "Y"
        contactObj.allowemail = "Y"
        contactObj.allowcall = "Y"

        Dim groups(0) As Integer
        groups(0) = "22299"

        Dim udf As New ContactSOAP.UDF
        udf.udfnameid = "985"
        udf.type = "Text"
        udf.data = "some value"
        Dim udfArray(0) As ContactSOAP.UDF
        udfArray(0) = udf

        Dim myContact As New ContactSOAP.ContactManager
        Dim contactID As Integer
contactID = myContact.createContact("username",
"password", contactObj, groups, udfArray)

        MsgBox("contactID: " &contactID)
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

$contact = array(
"uid" => "11d12",
            "title" => "Mr",
            "gender" => "M",
            "firstname" => "Joe",
            "lastname" => "Blogs",
            "buildingname" => "build name",
"buildingnumstreet" => "num street",
            "locality" => "locality",
            "city" => "city",
            "postcode" => "s11 7eb",
            "county" => "county",
            "country" => "UK",
            "membershipnumber" => "123456",
            "membershiptype" => "some type",
            "mobilenumber" => "077111111",
            "email" => "me@example.com",
            "dob" => "1973-02-16",
            "sourceid" => "116",
            "allowsms" => "Y",
            "allowcall" => "N",
            "allowemail" => "Y",
            "allowsnailmail" => "N");

$groups = array();
$groups = array(275,827);

$UDFs = array();
$UDFs[] = array("udfnameid"=>79, "data"=>"Test Co",
"type"=>"Text");
$UDFs[] = array("udfnameid"=>779, "data" =>"unique123",
"type"=>"Text");

$contactID = $client->createContact("username","password",
$contact,$groups,$UDFs);
        echo "contactID: ".$contactID;
```

RETURNS

```
contactID As Integer
```

2.1.2 Parameters: createContact

Field name	Max length	Type	Mandatory
title	-	Mr,Mrs,Ms,Dr,Miss,Prof	N
firstname	30	Alphanumeric	N
lastname	20	Alphanumeric	N
gender	1	M,F	N
dob	10	yyyy-mm-dd	N
email	50	Alphanumeric	N
mobilenumber	15	Numeric	N
landnumber	15	Numeric	N
sourceid	-	Numeric	**Y**
buildingname	30	Alphanumeric	N
buidlingnumstreet	30	Alphanumeric	N
locality	30	Alphanumeric	N
city	20	Alphanumeric	N
county	30	Alphanumeric	N
postcode	10	Alphanumeric	N
membershiptype	255	Alphanumeric	N
membershipnumber	255	Alphanumeric	N
allowsms	1	Y,N	N
allowcall	1	Y,N	N
allowemail	1	Y,N	N
allowsnailmail	1	Y,N	N
allowcall	1	Y,N	N
groupids	-	Array of Numeric	Y
udfs	-	Array of UDF	N

*** Note that sourceid and groupids are mandatory fields.*

Each method your customer can use to interact with PowerTexthas its own unique sourceid. This is known as a "Contact Interaction". This is useful for identifying interaction points if your data collection form is being used by more than one unit.

You can request your current sourceID's or set up a new one by emailing support@PowerText.co.uk.

Array of UDFs

Field name	Max length	Type	Mandatory
udfnameid	-	Numeric	N
Data	255	Alphanumeric	N
Type	-	Text, Date	N

UDF name ID's can be viewed via the PowerText console using the "UDF Manager" interface or listed by the SOAP API unitList() method.

Non mandatory special fields

The following “allow” fields must be set to one of “Y”, “N” or “”

```
allowsms          -   If set to Y the customer will receive SMS marketing.
allowemail        -   If set to Y the customer will receive Email marketing.
allowsnailmail    -   If set to Y the customer will receive Postal marketing.
allowcall         -   If set to Y the customer will receive Tele marketing.
```

If an “allow” field is set to “” (blank):-

If the contact already exists in the database then the stored contact allow field will not be overwritten.If there isn’t a stored contact then it will be created with N in the allow field.

2.2 Method: updateContact

This function can be used to update an existing contact record; you must however provide the unique PowerText contactid in the call.

2.2.1 Example: updateContact

Function:`updateContact (username As String, password As String, contactData As ContactSOAP.contactObject, groups() As Integer, udfs() As ContactSOAP.UDF ) As Integer`

.NET

```
    try
        Dim myService As New ContactSOAP.ContactManager
        Dim myContact As New ContactSOAP.contactObject

        ' change email address of given contactID
        myContact.contactid = "3794669"
        myContact.contactidSpecified = True
        myContact.email = "me@example.com"
        myContact.sourceid = "116"

        ' assign contact to this group, if not already a member
        Dim myGroups(0) As Integer
        myGroups(0) = "277"

        ' update or set udf value
        Dim udf As New ContactSOAP.UDF
        udf.udfnameid = "779"
        udf.type = "Text"
        udf.data = "unique3456"

        'add the UDF Object to an array of type UDF.
        Dim myUDFs(0) As ContactSOAP.UDF
        myUDFs(0) = udf

        Dim myResponse As Integer
        myResponse = myService.updateContact("username", "password",
 myContact, myGroups, myUDFs)
        'check the response for success
If myResponse.Equals(100) Then
            MsgBox("success: " & myResponse)
        Else
        MsgBox("success: " & myResponse)
        End If

    Catch ex As Exception
        MsgBox("updateContact () Exception: " + ex.Message)
    End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

$contact = array( "contactid" => 3794669,
                          "email" => "me@example.com",
                          "sourceid" => "116"
        );

        $groups = array();
        $groups = array(277);

        $UDFs = array();
        $UDFs[] = array("udfnameid" => 779,
                        "data" => "unique567",
                        "type" => "Text");

        $response = $client->updateContact("username","password",
                                           $contact,$groups,$UDFs);
        if($response == "100") {
            echo "Success";
        } else {
            echo "Failed: ".$response;
        }
```

RETURNS

```
statusID As Integer   (statusID = 100 for success)
```

2.2.2 Parameters: updateContact

```
see createContact 2.1.2
```

2.3 Method: getContact

This function can be used to obtain an existing contact record, you must however provide the unique PowerText contactID in the call.

2.3.1 Example: getContact

Function:`getContact (username As String, password As String, contactID As String) As ContactSOAP.contactStruct`

.NET

```
        Try

            Dim myService As New ContactSOAP.ContactManager
            Dim myContact As New ContactSOAP.contactStruct

            myContact = myService.getContact("username", "password", 3794669)

            If myContact.contactData Is Nothing Then
                MsgBox("getContact Failed to fetch ContactID 3794669")
            Else
                MsgBox("getContact Success, element email: " &
                                   myContact.contactData.email)
            End If

        Catch ex As Exception

            MsgBox("getContactExample() Exception: " + ex.Message)

        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

$contactID = 3794669;
$myContact = $client->getContact("username","password", $contactID);

if(!isset($myContact->contactData->contactid)) {
          echo "Failed: ".$myContact;
} else {
echo "Success, element email: ".$myContact->contactData->email;
}
```

RETURNS

```
object->contactid As Integer
object->uid As String
object->title As String
object->firstname As String
object->lastname As String
object->gender As String
object->dob As String
object->mobilenumber As String
object->landnumber As String
object->email As String
object->buildingname As String
object->buildingnumstreet As String
object->city As String
object->locality As String
object->county As String
object->postcode As String
object->country As String
object->membershiptype As String
object->membershipnumber As String
object->allowsms As String
object->allowemail As String
object->allowsnailmail As String
object->allowcall As String
object->sourceid As String
object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
object->groups As Array of Integers
```

2.4 Method: getContactEmail

This function can be used to find a contact record based on a fully qualified email address, wild card searches are not supported.

2.4.1 Example: getContactEmail

Function:getContactEmail (username As String, password As String, email As String) As ContactSOAP.contactStruct

.NET

```
        Try
            Dim myService As New ContactSOAP.ContactManager
            Dim myContact As New ContactSOAP.contactStruct

            myContact = myService.getContactEmail("username", "password",
 "me@example.com")

            If myContact.contactData Is Nothing Then
                MsgBox("getContactEmail Failed to find contact via email")
            Else
                MsgBox("getContactEmail Success, element contactID: " &
                                    myContact.contactData.contactid)
            End If
        Catch ex As Exception
            MsgBox("getContactEmail() Exception: " + ex.Message)
        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

$myContact = $client->getContactEmail("username","password",
"me@example.com");

if(!isset($myContact->contactData->contactid)) {
          echo "Failed: ".$myContact;
} else {
echo "Success, contactID: ".$myContact->contactData->contactid;
}
```

RETURNS

please refer to “getContact()” return specification.

2.5 Method: lookupContactByLastname

This function can be used to find contact record(s) with a specific surname, within a specified unitid, the surname must be passed in full; wildcard searches are not supported. This function also returns deleted contacts which can be deciphered via the value of the "deleted = Y/N" return data.

2.5.1 Example: lookupContactByLastname

Function:`lookupContactByLastname(username As String, password As String, unitid As Integer, lastname As String) As ContactSOAP.contactObjectExt()`

.NET

```
Try

            Dim myService As New ContactSOAP.ContactManager
            Dim myContact(0) As ContactSOAP.contactObjectExt

            myContact = myService.lookupContactByLastname("username",
 "password",
67,
"Wales")
            If myContact Is Nothing Then
                MsgBox("Failed to find contact via email")
            Else
                MsgBox("Success, first contactID found: " &
                       myContact(0).contactid)
            End If

Catch ex As Exception

            MsgBox("lookupContactByLastname() Exception: " + ex.Message)

        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);
$myContacts = $client->lookupContactByLastname("username",
"password",
                                                       67,
"Wales");
if(is_array($myContacts) && !empty($myContacts)){
          $firstContact = $myContacts[0];
 echo "Success, first contactID found: ".$firstContact->contactid;
} elseif(!is_array($myContacts)) {
echo "Failed: ".$myContact;
} else { echo "No Results."; }
```

RETURNS

```
Array of Object (
    object->contactid As Integer
    object->uid As String
    object->title As String
    object->firstname As String
    object->lastname As String
    object->gender As String
    object->dob As String
    object->mobilenumber As String
    object->landnumber As String
    object->email As String
    object->buildingname As String
    object->buildingnumstreet As String
    object->city As String
    object->locality As String
    object->county As String
    object->postcode As String
    object->country As String
    object->membershiptype As String
    object->membershipnumber As String
    object->allowsms As String
    object->allowemail As String
    object->allowsnailmail As String
    object->allowcall As String
    object->sourceid As String
    object->deleted As String
)
```

2.6 Method: lookupContactByUDF

This function can only be used on UDF fields which are configured to store unique values like a membership number, staff identifier or other user defined unique data sets.

2.6.1 Example: lookupContactByUDF

Function:`lookupContactByUDF (username As String, password As String, udfnameid As Integer, udfvalue As String) As Integer`

.NET

```
    Try
        Dim myService As New ContactSOAP.ContactManager
        Dim myContactID As Integer

        myContactID = myService.lookupContactByUDF("username",
                                  "password", 779, "unique567")
        If myContactID > 1 Then
            MsgBox("Success, found contactID: " &myContactID)
        Else
            MsgBox("Failed to find contact via UDF")
        End If
    Catch ex As Exception
        MsgBox("lookupContactByUDF() Exception: " + ex.Message)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
    $client = new SoapClient($wsdl);

    $myContactID = $client->lookupContactByUDF("username", "password",
                                               779, "unique567");

    if(is_numeric($myContactID) && $myContactID >= "1") {
        echo "Success, contactID: ".$myContactID;
    } else {
        echo "Failed: ".$myContactID;
    }
```

RETURNS

```
ContactID As Integer
```

2.7 Method: unsubscribe Contact

This function can be used to unsubscribe an email address and/or mobile number from future broadcasts. Pass mobile number(s)and matching contact records“Allow SMS” value(s)are set to “N”. Pass email address(s) and matching contact records “Allow Email” value(s) are set to “N”.

2.7.1 Example: unsubscribeContact

Function:unsubscribeContact (unsubscribeContact(username As String, password As String, unsubscribeContactRequest() As String) As String

.NET

```
        Try
            Dim myService As New ContactSOAP.ContactManager
            Dim myResponse As Integer

            Dim UnsubThese(1) As String
            UnsubThese(0) = "447911111111"
            UnsubThese(1) = "me@example.com"

            myResponse = myService.unsubscribeContact("username", "password",
                                                      UnsubThese)
            If myResponse.Equals(100) Then
                MsgBox("unsubscribeContact Success")
            Else
                MsgBox("unsubscribeContact Failed!")
            End If
        Catch ex As Exception
            MsgBox("unsubscribeContact() Exception: " + ex.Message)
        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);
        $UnsubThese = array("447911111111","me@example.com");
        $myResponse = $client->unsubscribeContact("username", "password",
$UnsubThese);
        if($myResponse == 100) {
            echo "unsubscribeContact Success";
        } else {
            echo "Failed: ".print_r($myResponse,1);
        }
```

RETURNS

```
statusID As Integer   (statusID = 100 for success)
```

2.8 Method: unsubscribeContactGroup

This function can be used to unsubscribe a contact from a specific groupID. Unsubscribing from a group will remove the contact from that group. You cannot unsubscribe a contact from all groups;they must remain in at least one group at all times. Use unsubscribeContact to remove a contact.

2.8.1 Example: unsubscribeContactGroup

Function:`unsubscribeContactGroup(username As String, password As String, contactid As Integer, groupid As Integer) As String`

.NET

```
        Try

            Dim myService As New ContactSOAP.ContactManager
            Dim myResponse As Integer

            myResponse = myService.unsubscribeContactGroup("username",
"password",
3794669,
                                                            827)
            If myResponse.Equals(100) Then
                MsgBox("unsubscribeContactGroup Success")
            Else
                MsgBox("unsubscribeContactGroup Failed!")
            End If
        Catch ex As Exception
            MsgBox("unsubscribeContactGroup() Exception: " + ex.Message)
        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

        $myResponse = $client->unsubscribeContactGroup("username",
                                        "password", 3794669, 827);

        if(is_numeric($myResponse) && $myResponse == 100) {
            echo "unsubscribeContactGroup Success";
        } else {
            echo "Failed: ".print_r($myResponse,1);
        }
```

RETURNS

```
statusID As Integer   (statusID = 100 for success)
```

2.9 Method: getUDFValue

This function can be used to load a specific contacts UDFID value for any given PowerText contactID.

2.9.1 Example: getUDFValue

Function:`getUDFValue(username As String, password As String, contactid As Integer, udfnameid As Integer) As String`

.NET

```
        Try
            Dim myService As New ContactSOAP.ContactManager
            Dim myResponse As String

            myResponse = myService.getUDFValue("username", "password",
                                               3794669, 779)

            If myResponse.Length() >= 1 Then
                MsgBox("getUDFValue Success: " + myResponse)
            Else
                MsgBox("getUDFValue Contact 3794669, UDFID 779, is blank")
            End If
        Catch ex As Exception
            MsgBox("getUDFValue() Exception: " + ex.Message)
        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

        $myResponse = $client->getUDFValue("username", "password",
                                           3794669, 779);

        if($myResponse instanceof SoapFault) {
            echo "Exception: ".$myResponse->getMessage();
        } elseif(strlen($myResponse) >= "1") {
            echo "Success, UDFValue: ".$myResponse;
        } else {
            echo "Empty, UDFValue: ".$myResponse;
        }
```

RETURNS

```
UDFValue As String
```

2.10 Method: setUDFValue

This function can be used to set a specific contact UDFID value for any given PowerText contactID.

2.10.1 Example: setUDFValue

Function:`setUDFValue(username As String, password As String, contactid As Integer, udfnameid As Integer, value As String, sourceid As Integer) As Integer`

.NET

```
Try
    Dim myService As New ContactSOAP.ContactManager
    Dim myResponse As Integer

    myResponse = myService.setUDFValue("username", "password",
                                  3794669, 779, "unique890", 116)

    If myResponse.Equals(100) Then
        MsgBox("setUDFValue Success: " + myResponse)
    Else
        MsgBox("setUDFValue failed: " + myResponse)
    End If
Catch ex As Exception
    MsgBox("setUDFValue() Exception: " + ex.Message)
End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
$client = new SoapClient($wsdl);

$myResponse = $client->setUDFValue("username", "password",
                                   3794669, 779, "unique890", 116);

if($myResponse instanceof SoapFault) {
    echo "Exception: ".$myResponse->getMessage();
} elseif($myResponse === "100") {
    echo "Success, setUDFValue: ".$myResponse;
} else {
    echo "Failed to setUDFValue:  ".$myResponse;
}
```

RETURNS

```
statusID As Integer   (statusID = 100 for success)
```

2.11 Method: getInteractionsInMonitoredGroup

This function can be used to load all interactions that have occurred within a monitored groupID.

2.11.1 Example: getInteractionsInMonitoredGroup

Function:`getInteractionsInMonitoredGroup(username As String, password As String, groupID As Integer) As ContactSOAP.interaction()`

.NET

```
        Try

            Dim myService As New ContactSOAP.ContactManager
            Dim myResponse(0) As ContactSOAP.interaction

            myResponse = myService.getInteractionsInMonitoredGroup(
                                        "username", "password", 275)

            If myResponse Is Nothing Then
                MsgBox("returned nothing")
            Else
                MsgBox("Success: " + myResponse(0).contactid)
            End If

        Catch ex As Exception

            MsgBox("getInteractionsInMonitoredGroup Exception: " + ex.Message)

        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

        $myResponse = $client->getInteractionsInMonitoredGroup(
                                        "username", "password", 275);
        if($myResponse instanceof SoapFault) {
            echo "Exception: ".$myResponse->getMessage();
        } elseif(is_array($myResponse) && !empty($myResponse)) {
foreach($myResponse as $record) {
                echo "<pre>".print_r($record,1)."</pre>\n";
            }
        } else {
echo "Empty Response!";
        }
```

RETURNS

```
Array of Object (
object->recordid As String
object->accountidAs Integer
object->sourcename As String
object->unitidAs Integer
object->unitname As String
object->groupidAs Integer
object->groupname As String
object->contactidAs Integer
object->contactgender As String
object->contactfirstname As String
object->contactlastname As String
object->contactmobilenumber As String
    object->contactlandnumber As String
object->contactemailaddress As String
object->contactdateofbirth As String
object->contactaddressline1 As String
object->contactaddressline2 As String
object->contactaddresslocality As String
object->contactaddresscity As String
object->contactaddresscounty As String
object->contactaddresspostcode As String
object->contactaddresscountry As String
object->contactcreatedtime As String
object->contactdeleted As String
object->contactuid As String
object->contactlastupdatetime As String
object->contactmembershipnumber As String
object->contactmembershiptype As String
object->contactoutwardcode As String
object->contactinwardcode As String
object->contactallowsms As String
object->contactallowemail As String
object->contactallowsnailmail As String
    object->contactallowcall As String
object->contactdeadmobile As String
object->contactdeademail As String
object->contactactiveemail As String
object->contactdeadmobiletime As String
object->contactlastfailedsms As String
object->contactdeadmobilecount As String
object->contactunsubsmstime As String
object->contactunsubemailtime As String
object->contactfailedcount As String
object->contactemailhardbounce As String
object->contactemailsoftbounce As String
object->contactdeademailtime As String
object->smsdata As String
object->timestamp As String
object->contactudfs As Array of Objects (o->udfnameid, o->data, o->type))
```

2.12 Method: deleteInteractionsInMonitoredGroup

This function can be used to remove a record(s) from the monitored group interactions log.

2.12.1 Example: deleteInteractionsInMonitoredGroup

Function:`deleteInteractionsInMonitoredGroup(username As String, password As String, recordID() As Integer) As Integer`

.NET

```
        Try

            Dim myService As New ContactSOAP.ContactManager
            Dim myResponse As Integer
            Dim myRecords(0) As Integer
            myRecords(0) = 12345

            myResponse = myService.deleteInteractionsInMonitoredGroup(
                                   "username", "password", myRecords)

            If myResponse.Equals(100) Then
                MsgBox("deleteInteractionsInMonitoredGroup success")
            Else
                MsgBox("deleteInteractionsInMonitoredGroup failed: " &
                       myResponse)
            End If
        Catch ex As Exception
            MsgBox("deleteInteractionsInMonitoredGroup Exception: " +
                   ex.Message)
        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Contact.wsdl";
        $client = new SoapClient($wsdl);

        $myResponse = $client->deleteInteractionsInMonitoredGroup(
                          "username", "password", array("12345"));

        if($myResponse instanceof SoapFault) {
            echo "Exception: ".$myResponse->getMessage();
        } elseif($myResponse == "100") {
            echo "success"
        } else {
echo "failed: ".$myResponse;
        }
```

RETURNS

```
statusID As Integer   (statusID = 100 for success)
```

3. Statistics API

The Statistics API provides methods for accessing cached statistics data, live unit lists and live group lists.The statistical cache data is automatically updated every 24hrs.

File:https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl**Namespace:**StatSOAP

3.1 Method: ageProfile

This function can be used to obtain a list of contact ages and the number of contacts in each age.

3.1.1 Example: ageProfile

Function:`ageProfile(username As String, password As String, unitID As Integer) As StatSOAP.Age()`

.NET

```
   Try
        Dim myService As New StatSOAP.Stat
        Dim myResponse(0) As StatSOAP.Age
        myResponse = myService.ageProfile("username", "password", 67)
        If myResponse Is Nothing Then
            MsgBox("returned nothing")
        Else
            MsgBox("first returned row, count element: " & myResponse(0).count)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);
    $myResponse = $client->ageProfile("username", "password", 67);
    if($myResponse instanceof SoapFault) {
        echo "Exception: ".$myResponse->getMessage();
    } elseif(is_array($myResponse) && !empty($myResponse)) {
        foreach($myResponse as $record)
            echo "<pre>".print_r($record,1)."</pre>\n";
    } else {
        echo "Empty Response!";
    }
```

RETURNS

```
Array of Object ( object->yearsold As Integer, object->count As Integer)
```

3.2 Method: birthdayProfile

This function can be used to obtain a list of months and number of contacts born in each month.

3.2.1 Example: birthdayProfile

Function:birthdayProfile(username As String, password As String, unitID As Integer) As StatSOAP.Month()

.NET

```
   Try
        Dim myService As New StatSOAP.Stat
        Dim myResponse(0) As StatSOAP.Month

        myResponse = myService.birthdayProfile("username", "password", 67)

        If myResponse Is Nothing Then
            MsgBox("returned nothing")
        Else
            MsgBox("first returned row, count element: " & myResponse(0).count)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);

    $myResponse = $client->birthdayProfile("username", "password", 67);

    if($myResponse instanceof SoapFault) {
        echo "Exception: ".$myResponse->getMessage();
    } elseif(is_array($myResponse) && !empty($myResponse)) {
        foreach($myResponse as $record) {
            echo "<pre>".print_r($record,1)."</pre>\n";
        }
    } else {
        echo "Empty Response!";
    }
```

RETURNS

```
Array of Object ( object->name As String, object->count As Integer)
```

3.3 Method: groupList

This function can be used to load a list of active groups belonging to a given unitID.

3.3.1 Example: groupList

Function:`groupList(username As String, password As String, unitID As Integer) As StatSOAP.Folder()`

.NET

```
    Try
        Dim myService As New StatSOAP.Stat
        Dim myResponse(0) As StatSOAP.Folder

        myResponse = myService.groupList("username", "password", 67)

        If myResponse Is Nothing Then
            MsgBox("returned nothing")
        Else
            MsgBox("first returned row, foldername element: " +
                                          myResponse(0).foldername)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Messa
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);

    $myResponse = $client->groupList("username", "password", 67);

    if($myResponse instanceof SoapFault) {
        echo "Exception: ".$myResponse->getMessage();
    } elseif(is_array($myResponse) && !empty($myResponse)) {
        foreach($myResponse as $record) {
            echo "<pre>".print_r($record,1)."</pre>\n";
        }
    } else {
        echo "Empty Response!";
    }
```

RETURNS

```
Array of Object (
    object->foldername As String,
    object->groups As Array of Object (o->groupid, o->groupname)
)
```

3.4 Method: groupStats

This function can be used to load group statistical data from groups associated with your account.

3.4.1 Example: groupStats

Function:`groupStats(username As String, password As String, unitID As Integer) As StatSOAP.GroupStats()`

.NET

```
    Try

        Dim myService As New StatSOAP.Stat
        Dim myResponse(0) As StatSOAP.GroupStats

        myResponse = myService.groupStats("username", "password", 67)

        If myResponse Is Nothing Then
            MsgBox("returned nothing")
        Else
            MsgBox("first returned row, name element: " + myResponse(0).name)
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message
End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);

    $myResponse = $client->groupStats("username", "password", 67);
    if($myResponse instanceof SoapFault) {
        echo "Exception: ".$myResponse->getMessage();
    } elseif(is_array($myResponse) && !empty($myResponse)) {
        foreach($myResponse as $record)
            echo "<pre>".print_r($record,1)."</pre>\n";
    } else {
        echo "Empty Response!";
    }
```

RETURNS

```
Array of Object (
    object->groupID As Integer,
    object->name As String,
    object->stats As Array of Object (o->activemobilescached,
        o->activeemailscached,o->activemailerscached,o->contactscached)
)
```

3.5 Method: joinersByPeriod

This function can be used to load statistical data detailing the number of joiners in the last week and month for a given unitID.

3.5.1 Example: joinersByPeriod

Function:`joinersByPeriod(username As String, password As String, unitID As Integer) As StatSOAP.Period()`

.NET

```
    Try
        Dim myService As New StatSOAP.Stat
        Dim myResponse(0) As StatSOAP.Period

        myResponse = myService.joinersByPeriod("username", "password", 67)

        If myResponse Is Nothing Then
            MsgBox("returned nothing")
        Else
            MsgBox("first returned row, name element: " + myResponse(0).name)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Message
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);
    $myResponse = $client->joinersByPeriod("username", "password", 67);

    if($myResponse instanceof SoapFault) {
        echo "Exception: ".$myResponse->getMessage();
    } elseif(is_array($myResponse) && !empty($myResponse)) {
        foreach($myResponse as $record)
            echo "<pre>".print_r($record,1)."</pre>\n";
    } else {
        echo "Empty Response!";
    }
```

RETURNS

```
Array of Object (object->name, object->count)
```

3.6 Method: previousBroadcastList

This function can be used to load a list of all previous broadcasts that have been made from your account.

3.6.1 Example: previousBroadcastList

Function:`previousBroadcastList(username As String, password As String, unitID As Integer) As StatSOAP.BroadcastsArray`

.NET

```
    Try
        Dim myService As New StatSOAP.Stat
        Dim myResponse As StatSOAP.BroadcastsArray

        myResponse = myService.previousBroadcastList("username",
                                                     "password", 67)

        If myResponse Is Nothing Then
            MsgBox("returned nothing")
        Else
            MsgBox("first row eflyer name: " + myResponse.eflyers(0).name)
            MsgBox("first row eflyer name: " + myResponse.tflyers(0).name)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);
    $myResponse = $client->previousBroadcastList("username", "password", 67);

    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(isset($myResponse->eflyers) && isset($myResponse->tflyers)) {
        echo "<pre>this->eflyers: ".print_r($myResponse->eflyers,1)."</pre>";
        echo "<pre>this->tflyers: ".print_r($myResponse->tflyers,1)."</pre>";
    } else {
        echo "Empty Response!";
    }
```

RETURNS

```
Array of Object (object->queueid As Integer, object->qty As Integer,
                 object->name As String, object->sentdate As String )
```

3.7 Method: refreshCache

This function can be used to refresh your statistical data cache. It can only be called once in any given 24hr period. If you call this function more than once in 24hrs you will receive a SoapFault.

3.7.1 Example: refreshCache

Function:`refreshCache(username As String, password As String) As Integer`

.NET

```
    Try
        Dim myService As New StatSOAP.Stat
        Dim myResponse As Integer

        myResponse = myService.refreshCache("username", "password")

        If myResponse.Equals(100) Then
            MsgBox("success")
        Else
            MsgBox("failed " + myResponse)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);

    $myResponse = $client->refreshCache("username", "password");

    if($myResponse instanceof SoapFault) {
        die("SoapFault: ".$myResponse->getMessage());
    } elseif($myResponse == '100') {
        echo "Success!";
    } else {
        echo "Failed! ".$myResponse;
    }
```

RETURNS

```
statusID As Integer   (statusID = 100 for success)
```

3.8 Method: unitList

This function can be used to load a list of active units and their ID's from your account.

3.8.1 Example: unitList

Function:`unitList(username As String, password As String) As StatSOAP.Unit()`

.NET

```
    Try

        Dim myService As New StatSOAP.Stat
        Dim myResponse(0) As StatSOAP.Unit

        myResponse = myService.unitList("username", "password")

        If myResponse Is Nothing Then
            MsgBox("returned nothing")
        Else
            MsgBox("first row: name = " + myResponse(0).name)
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Stat.wsdl";
    $client = new SoapClient($wsdl);

    $myResponse = $client->unitList("username", "password");

    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse) && !empty($myResponse)) {
        foreach($myResponse as $unit) {
            echo "<pre>".print_r($unit,1)."</pre>";
        }
    } else {
        echo "Empty Response!";
    }
```

RETURNS

```
Array of Object ( object->unitid As Integer,
                  object->name As String,
                  object->isroot As String )
```

4. UDF Unit Query API

The UDF Unit Query API provides methods for searching contact records via UDF criteria.

File:https://secure.powertext.co.uk/SOAP/V2/UDFUnitQuery.wsdl**Namespace:**UDFQuerySOAP

4.1 Method: contactCheckUDF

Looks for a contact matching the given emailaddress and within the given UnitID, checks the matching contact has a UDFID value equal to that of UDFVAL parameter.

4.1.1 Example: contactCheckUDF

Function: `contactCheckUDF(username As String, password As String, unitid As Integer, udfid As Integer, udfval As String, emailaddress As String) As String`

.NET

```
    Try
        Dim myService As New UDFQuerySOAP.UDFUnitQuery
        Dim myResponse As String
        myResponse = myService.contactCheckUDF("username", "password", 67,
                                        779, "unique891", "me@example.com")
        If myResponse Is Nothing Then
            MsgBox("bad response")
        ElseIf myResponse = "true" Or myResponse = "false" Then
            MsgBox("contact matched query: " + myResponse)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UDFUnitQuery.wsdl";
    $client = new SoapClient($wsdl);
    $myResponse = $client->contactCheckUDF("username", "password", 67, 779,
 "unique891", "me@example.com");
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(in_array($myResponse, array('true','false'))) {
        echo "contact matched query: ".$myResponse;
    }
```

RETURNS

```
matchfound As String  (available responses: "true" or "false")
```

4.2 Method: getContactData

Looks for a contact matching the given emailaddress and within the given UnitID, checks the matching contact has a UDFID value equal to that of UDFVAL parameter.

4.2.1 Example: getContactData

Function: `getContactData(username As String, password As String, unitids As String, groupids As String, udfid As Integer, udfval As String) As UDFQuerySOAP.ContactData()`

.NET

```
    Try
        Dim myService As New UDFQuerySOAP.UDFUnitQuery
        Dim myResponse(0) As UDFQuerySOAP.ContactData

        myResponse = myService.getContactData("username", "password",
                                              "67,187", "275,827", 479, "test1")

        If myResponse Is Nothing Then
            MsgBox("bad response")
        Else
            MsgBox("first row contactid: " & myResponse(0).contactid)
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UDFUnitQuery.wsdl";
    $client = new SoapClient($wsdl);

    $myResponse = $client->getContactData("username", "password",
                                          "67,187", "275,827", 479, "test1");

    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
    object->unitname As String
    object->unitid As Integer
    object->groupid As Integer
    object->groupname As String
    object->contactid As Integer
    object->uid As String
    object->gender As String
    object->title As String
    object->firstname As String
    object->lastname As String
    object->dob As String
    object->buildingname As String
    object->buildingnumstreet As String
    object->locality As String
    object->postcode As String
    object->city As String
    object->country As String
    object->county As String
    object->mobilenumber As String
    object->landnumber As String
    object->email As String
    object->createdtime As String
    object->lastupdatetime As String
    object->allowsms As String
    object->allowemail As String
    object->allowsnailmail As String
    object->allowcall As String
    object->deadmobile As String
    object->deademail As String
)
```

4.3 Method: udfList

This function can be used to load a list of UDF’s and their configuration data from your account.

4.3.1 Example: udfList

Function: `udfList(username As String, password As String, unitid As Integer) As UDFQuerySOAP.UDFData()`

.NET

```
    Try
        Dim myService As New UDFQuerySOAP.UDFUnitQuery
        Dim myResponse(0) As UDFQuerySOAP.UDFData

        myResponse = myService.udfList("username", "password", 67)

        If myResponse Is Nothing Then
            MsgBox("empty response")
        Else
            MsgBox("first row udfid: " & myResponse(0).udfid)
        End If
    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UDFUnitQuery.wsdl";
    $client = new SoapClient($wsdl);

    $myResponse = $client->udfList("username", "password", 67);

    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
    object->udfid As Integer,
    object->udfname As String,
    object->udftype As String,
    object->udfoptions As Array of String
)
```

5. Broadcast API

The Broadcast API provides methods for rescheduling of previous broadcasts.

File:https://secure.powertext.co.uk/SOAP/V2/Broadcast.wsdl**Namespace:**BroadcastSOAP

5.1 Method: resendEflyer

Can be used to send a previous Eflyer broadcast to new recipients and a different from address.

5.1.1 Example: resendEflyer

Function: `resendEflyer(username As String, password As String, broadcastID As Integer, fromAddress As String, recipients() As String, shareUID As String) As Integer`

.NET

```
    Try
        Dim myService As New BroadcastSOAP.Broadcast
        Dim myRecipients(0) As String
        Dim myResponse As Integer
        myRecipients(0) = "me@example.com"
        myRecipients(1) = "you@example.com"
        myResponse = myService.resendEflyer("username", "password", 440859,
"from@example.com", myRecipients, "387f0d146655a47c70b3387b0aad2f3a")
        If myResponse.Equals(100) Then
            MsgBox("success")
        Else
            MsgBox("failed " & myResponse)
        End If
    Catch ex As Exception
            MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Broadcast.wsdl";
    $client = new SoapClient($wsdl);
    $myResponse = $client->resendEflyer("username", "password", 440859,
        "from@example.com", array("to@example.com"),
"387f0d146655a47c70b3387b0aad2f3a");
    if($myResponse instanceof SoapFault) {
        die("SoapFault: ".$myResponse->getMessage());
    } elseif($myResponse == '100') {
        echo "Success!";
    } else {
        echo "Failed! ".$myResponse;
    }
```

```
RETURNSstatusID As Integer    (statusID = 100 for success)
```

5.2 Method: sendNewEflyer

Can be used to create and send an Eflyer / Email broadcast to new or existing recipient(s). You must supply the from email address, the recipient(s) email address(es), the subject and the content. Content must be provided in both html and plain text format. If the recipient email address is not in the database, a contact record will be created containing the new recipient email address.

5.2.1 Example: sendNewEflyer

Function: `sendNewEflyer(username As String, password As String, unitID As Integer, fromAddress As String, recipients() As String, subject As String, htmlContent As String, textContent As String) broadcastID As Integer`

.NET

```
    Try
        Dim myService As New BroadcastSOAP.Broadcast
        Dim myResponse As Integer
        Dim myRecipients(1) As String

        myRecipients(0) = "me@example.com"
        myRecipients(1) = "you@example.com"

        Dim unitID As Integer = 67

        Dim fromAddress As String = "from@example.com"

        Dim subject As String = "New SOAP API Test Message"

        Dim textContent As String = "This is a test message"

        Dim htmlContent As String = "<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML
1.0 Transitional//EN' 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd'><html xmlns='http://www.w3.org/1999/xhtml'><head><meta http-
equiv='Content-Type' content='text/html; charset=iso-8859-1' /><title>new
sendNewEflyer() message</title></head><body><h1>This is a test message.
</h1></body></html>"

        myResponse = myService.sendNewEflyer("username", "password", unitID,
                fromAddress, myRecipients, subject, htmlContent, textContent)

        If myResponse >= 1 Then
            MsgBox("success: BroadcastID " & myResponse)
        Else
            MsgBox("failed " & myResponse)
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.PowerText.co.uk/SOAP/V2/Broadcast.wsdl";
$client = new SoapClient($wsdl);

$htmlContent = "<html><body><h2>Test</h2><p>hello</p></body></html>";
$textContent = "Test\r\n\r\nhello";

$myResponse = $client->sendNewEflyer("username", "password", 67,
                                     "from@example.com",
                                     array("to@example.com"),$htmlContent,
                                     $textContent);

if($myResponse instanceof SoapFault) {
    die("SoapFault: ".$myResponse->getMessage());
} elseif($myResponse >= '1') {
    echo "Success, broadcastID: ".$myResponse;
} else {
    echo "Failed! ".$myResponse;
}
```

RETURNS

```
broadcastID As Integer
```

5.3 Method: sendNewTflyer

Can be used to create and send a new Tflyer / SMS broadcast to new or existing recipient(s). You must supply the recipient(s) mobile number(s) and the content of the message in plain text. If the recipient mobile number is not in the database, a contact record will be created containing the new recipient mobile number.

5.3.1 Example: sendNewTflyer

Function: `sendNewTflyer(username As String, password As String, unitID As Integer, recipients() As String, smsContent As String) broadcastID As Integer`

.NET

```
    Try

        Dim myService As New BroadcastSOAP.Broadcast
        Dim myRecipients(1) As String
        Dim myResponse As Integer
        myRecipients(0) = "447911111111"
        myRecipients(1) = "07911 111112"

        Dim unitID As Integer = 67
        Dim smsContent As String = "This is a test sms message from the new
sendNewTflyer() function!"

        myResponse = myService.sendNewTflyer("username", "password", unitID,
myRecipients, smsContent)

        If myResponse >= 1 Then
            MsgBox("success: BroadcastID " & myResponse)
        Else
            MsgBox("failed " & myResponse)
        End If

    Catch ex As Exception

        MsgBox("Exception: " + ex.Message)

    End Try
```

.PHP

```
    $wsdl = "https://secure.PowerText.co.uk/SOAP/V2/Broadcast.wsdl";
    $client = new SoapClient($wsdl);
    $smsContent = "Test hello";
    $myResponse = $client->sendNewTflyer("username", "password",
                                         array("447911111111","447911111112"),
                                         $smsContent);
    if($myResponse instanceof SoapFault) {
        die("SoapFault: ".$myResponse->getMessage());
    } elseif($myResponse >= '1') {
        echo "Success, broadcastID: ".$myResponse;
    } else {
        echo "Failed! ".$myResponse;
    }
```

RETURNS

```
broadcastID As Integer
```

5.4 Method: broadcastStatusTflyer

Can be used to obtain delivery status data of a previous broadcastID to a specified mobile number(s). This function is limited to a maximum of 250 recipients per call.

5.4.1 Example: broadcastStatusTflyer

Function: `broadcastStatusTflyer(username As String, password As String, broadcastID As Integer, recipients() As String) As BroadcastSOAP.tflyerStatus()`

.NET

```
Try
    Dim myService As New BroadcastSOAP.Broadcast
    Dim broadcastID As Integer
    Dim myRecipients(0) As String
    Dim myResponse(0) As BroadcastSOAP.tflyerStatus
    myRecipients(0) = "447711111111"
    myRecipients(1) = "447711111112"
    broadcastID = 12345
    myResponse = myService.broadcastStatusTflyer("username", "password",
                                                 broadcastID,myRecipients)
    If myResponse Is Nothing Then
        MsgBox("empty response")
    Else
        MsgBox("first row: " & myResponse(0).mobilenumber)
    End If
Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$wsdl = "https://secure.PowerText.co.uk/SOAP/V2/Broadcast.wsdl";
$client = new SoapClient($wsdl);

$myResponse = $client->broadcastStatusTflyer("username", "password",
                          12345, array("4477111111","447711111112"));

if($myResponse instanceof SoapFault) {
    die("Exception: ".$myResponse->getMessage());
} elseif(is_array($myResponse)) {
    echo "<pre>response: ".print_r($myResponse,1)."</pre>";
} else {
    echo "bad response: ".$myResponse;
}
```

RETURNS

```
Array of Object (
    object->contactid As Integer,
    object->mobilenumber As String,
    object->status As String,
    object->clicks As Integer
)
```

5.5 Method: broadcastStatusEflyer

Can be used to obtain delivery status data of a previous broadcastID to a specified email address(es). This function is limited to a maximum of 250 recipients per call.

5.5.1 Example: broadcastStatusEflyer

Function: `broadcastStatusEflyer(username As String, password As String, broadcastID As Integer, recipients() As String) As BroadcastSOAP.eflyerStatus()`

.NET

```
    Try
        Dim myService As New BroadcastSOAP.Broadcast
        Dim broadcastID As Integer
        Dim myRecipients(0) As String
        Dim myResponse(0) As BroadcastSOAP.eflyerStatus
        myRecipients(0) = "me@example.com"
        myRecipients(1) = "you@example.com"
        broadcastID = 67890
        myResponse = myService.broadcastStatusEflyer("username", "password",
                                                      broadcastID,myRecipients)
        If myResponse Is Nothing Then
            MsgBox("empty response")
        Else
            MsgBox("first row: " & myResponse(0).email)
        End If
    Catch ex As Exception
            MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
    $wsdl = "https://secure.PowerText.co.uk/SOAP/V2/Broadcast.wsdl";
    $client = new SoapClient($wsdl);
    $myResponse = $client->broadcastStatusEflyer("username", "password",
                                   67890,
array("me@example.com","you@example.com"));

    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
    object->contactid As Integer,
    object->email As String,
    object->status As String,
    object->reads As Integer,
    object->clicks As Integer,
)
```

5.6 Method: sendForcedEflyer

Can be used to create and send an Eflyer / Email broadcast to new or existing recipient(s) even if deleted or opted out. You must supply the recipient(s) email address(es) and an Eflyer ID. If the recipient email address is not in the database, a contact record will be created containing the new recipient email address but will be marked as deleted. This Eflyer will go to recipients regardless of them being deleted or having unsubscribed.

5.6.1 Example: sendForcedEflyer

Function: `sendForcedEflyer(username As String, password As String, eflyerID As Integer, recipients() As String) broadcastID As Integer`

.NET

```
    Try
        Dim myService As New BroadcastSOAP.Broadcast
        Dim myResponse As Integer
        Dim myRecipients(1) As String

        myRecipients(0) = "me@example.com"
        myRecipients(1) = "you@example.com"

        Dim eflyerID As Integer = 67

        myResponse = myService.sendForcedEflyer("username", "password",
eflyerID, myRecipients)

        If myResponse >= 1 Then
            MsgBox("success: BroadcastID " & myResponse)
        Else
            MsgBox("failed " & myResponse)
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.PowerText.co.uk/SOAP/V2/Broadcast.wsdl";
$client = new SoapClient($wsdl);

$myResponse = $client->sendForcedEflyer("username", "password", 123456,
                                     array("to@example.com"));

if($myResponse instanceof SoapFault) {
    die("SoapFault: ".$myResponse->getMessage());
} elseif($myResponse >= '1') {
    echo "Success, broadcastID: ".$myResponse;
} else {
    echo "Failed! ".$myResponse;
}
```

RETURNS

```
broadcastID As Integer
```

6. Bookings API

The bookings api allows for creation, updating and retrieval of booking specific data.

File:https://secure.powertext.co.uk/SOAP/V2/Bookings.wsdl**Namespace:** BookingsSOAP

6.1 Method: createBooking

This function can be used to add a new booking record, it will return the provided bookingID on success. Before calling the function we must have a valid contact_id – if this is a new contact the createContact must be called first to obtain a new contact_id.

6.1.1 Example: createBooking

Function:`createBooking(username As String, password As String, booking As BookingSOAP.bookingObject, booking_areas() As Integer) booking_id As Integer`

.NET

```
Try

    Dim myService As New BookingSOAP.BookingManager
 Dim myBooking As New BookingSOAP.bookingObject
    Dim myBookingAreas(1) As Integer
    Dim myBookingID As Integer

    myBookingAreas(0) = "101" ' reserve area 101 and 102"
    myBookingAreas(1) = "102" ' reserve area 101 and 102

    myBooking.booking_contact_id = "5282391"
myBooking.booking_contact_idSpecified = True
myBooking.booking_enquiry_source_id = "2"
myBooking.booking_enquiry_source_idSpecified = True
    myBooking.booking_party_size = "10"
    myBooking.booking_party_sizeSpecified = True
    myBooking.booking_stage_value = "1"
    myBooking.booking_stage_valueSpecified = True
    myBooking.booking_type_id = "2"
    myBooking.booking_type_idSpecified = True
    myBooking.booking_unit_id = "67"
    myBooking.booking_unit_idSpecified = True
    myBooking.booking_deposit_paid = 0
    myBooking.booking_deposit_paidSpecified = True
    myBooking.booking_hpbr_drink = 0
    myBooking.booking_hpbr_drinkSpecified = True
   myBooking.booking_hpbr_food = 0
    myBooking.booking_hpbr_foodSpecified = True
   myBooking.booking_hpbr_entertainment = 0
    myBooking.booking_hpbr_entertainmentSpecified = True
    myBooking.booking_party_datetime = "2013-05-02 20:30:00"
    myBooking.booking_enquiry_datetime = "2013-03-21 21:00:06"
    myBooking.booking_data_element_03 = "beer"
    myBooking.booking_data_element_20 = "steak"

    myBookingID = myService.createBooking("username", "password",
myBooking, myBookingAreas)
    If myBookingID >= 1 Then
        MsgBox("Success, booking_id: " & myBookingID)
    Else
        MsgBox("Failed to create booking.")
    End If

Catch ex As Exception
    MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl",array("trace"=>1,"exceptions"=>0));

$booking = array(
   'booking_contact_id'=>'5282391',
   'booking_enquiry_source_id'=>'2',
   'booking_party_size'=> '10',
   'booking_stage_value'=> '1',
   'booking_type_id'=> '2',
   'booking_unit_id'=> '67',
   'booking_party_datetime'=> "2013-05-02 20:30:00",
   'booking_enquiry_datetime'=> "2013-03-21 21:00:06",
  'booking_deposit_paid'=> '0',
   'booking_hpbr_drink'=> '0',
   'booking_hpbr_food'=> '0',
   'booking_hpbr_entertainment'=> '0',
   'booking_data_element_03'=> "wine",
   'booking_data_element_20'=> "chicken" );

$booking_areas = array('108','109');

$myBookingID = $client->createBooking("username","password",
                                       $booking,$booking_areas);

if($myBookingID instanceof SoapFault || $myBookingID < '1') {
    echo "SOAP Fault: ".$myBookingID->faultstring;
} else {
    echo "Success, new booking_id: ".$myBookingID;
}
```

RETURNS

```
booking_id As Integer
```

6.1.2 Booking Stage Values

Booking stage field can contain “any” numeric value between 1 and 600. PowerText have pre-defined status names assigned to different numeric values, as shown below. It is advised to use the status definitions as are, where possible, but it is possible to use extra stages inside external applications to facilitate between stage workflow.

```
    New (10)  range (1-19)
    Customer Contacted (20)  range (20-29)
  Info Sent (30)  range (30-39)
  Follow-up (40)  range (40-49)
    Confirmation Received (50)  range (50-59)
    Deposit Secured (60)  range (60-69)
Complete (100)  range (100-499)
    Unsuccessful (500)  range (500-549)
    Cancelled (550)  range (550-600)
```

When using bespoke stages the following rules will apply within PowerText. If a value between 1-19 is given, this will be treated by PowerText as “New”. You can view the range of acceptance in the above list.

6.2 Method: createBookingNote

This function can be used to add a note(s) to a booking record. Multiple notes may be added to a booking_id to form a thread of notes against said booking_id. To create a customer comment, use a pt_user_id value of “0”.

6.2.1 Example: createBookingNote

Function:`createBookingNoteAsync(username As String, password As String, booking_note As BookingSOAP.bookingNoteObject) booking_note_id As Integer`

.NET

```
Try
    Dim myService As New BookingSOAP.BookingManager
    Dim myBookingNote As New BookingSOAP.bookingNoteObject
    Dim myBookingNoteID As Integer
    myBookingNote.booking_note_booking_id = 4
    myBookingNote.booking_note_booking_idSpecified = True
    myBookingNote.booking_note_pt_user_id = 40
myBookingNote.booking_note_pt_user_idSpecified = True
    myBookingNote.booking_note_text = "This is a test note on bookingID 4"
    myBookingNote.booking_note_created_datetime = "2013-03-22 16:16:28"
    myBookingNoteID = myService.createBookingNote("username", "password",
                                                  myBookingNote)
   If myBookingNoteID >= 1 Then
        MsgBox("Success, booking_note_id: " & myBookingNoteID)
    Else
        MsgBox("Failed to create booking note.")
End If
Catch ex As Exception
    MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl",array("trace"=>1,"exceptions"=>0));
$booking_note = array(
'booking_note_booking_id'=>'5',
        'booking_note_pt_user_id'=>'40',
        'booking_note_text'=>'Test note on bookingID 5',
        'booking_note_created_datetime'=>'2013-03-25 12:52:11' );
$myBookingNoteID = $client->createBookingNote("username","password",
                                              $booking_note);
if($myBookingNoteID instanceof SoapFault || $myBookingNoteID < 1) {
    echo "SOAP Fault: ".$myBookingNoteID->faultstring;
} else {
    echo "Failed to create booking note";
}
```

RETURNS

```
booking_note_id As Integer
```

6.3 Method: updateBooking

This function can be used to update an existing booking record. It will return 100 on success.

6.3.1 Example: updateBooking

Function:`updateBooking(username As String, password As String, booking As BookingSOAP.bookingObject, booking_areas() As Integer) status As Integer`

.NET

```
Try

    Dim myService As New BookingSOAP.BookingManager
 Dim myBooking As New BookingSOAP.bookingStruct
    Dim myBookingAreas(2) As Integer
   Dim mySuccess As New Integer

'load a previous booking object
    myBooking = myService.getBooking("username","password", 4)

    'change the booking_stage_value to 10
    myBooking.booking.booking_stage_value = 10

    'change the booking areas
    myBookingAreas(0) = 98
    myBookingAreas(1) = 101
    myBookingAreas(2) = 102
    myBooking.booking_areas = myBookingAreas

    'add / change some extended booking data fields
    myBooking.booking.booking_data_element_03 = "wine"
    myBooking.booking.booking_data_element_20 = "" ' clear previous value

    'commit changes to database
    mySuccess = myService.updateBooking("username","password",
myBooking.booking, myBooking.booking_areas)

  If mySuccess.Equals(100) Then
        MsgBox("Successfully updated booking")
    Else
        MsgBox("Failed to get booking.")
    End If

Catch ex As Exception

    MsgBox("Exception: " + ex.Message)

End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl", array("trace"=>1,"exceptions"=>0));

// load bookingID 5 data
$bookingStruct = $client->getBooking("username","password",5);

// update some of the loaded values
$bookingStruct->booking->booking_stage_value = '10';
$bookingStruct->booking->booking_data_element_03 = 'cocktails';

// clear previous value
$bookingStruct->booking->booking_data_element_20 = '';

// reserve a different set of areas
$bookingStruct->booking_areas = array('98','99','101');

// attempt to commit changes
$myBookingID = $client->updateBooking("username","password",
$bookingStruct->booking,
                                      $bookingStruct->booking_areas);

if($myBookingID instanceof SoapFault) {
    echo "SOAP Fault: ".$myBookingID->faultstring;
} elseif($myBookingID == '100') {
    echo "Successfully updated booking record";
} else {
    echo "Failed to update booking";
}
```

RETURNS

```
100 As Integer
```

6.4 Method: getBooking

This function can be used to fetch a single,specific, booking record.

6.4.1 Example: getBooking

Function:`getBooking(username As String, password As String, booking_id As Integer) As BookingSOAP.bookingStruct`

.NET

```
Try
    Dim myService As New BookingSOAP.BookingManager
    Dim myBooking As New BookingSOAP.bookingStruct

    myBooking = myService.getBooking("username","password", 5)

    If myBooking.booking Is Nothing Then

        MsgBox("Failed to get booking.")

    Else

        MsgBox("Success, booking_contact_id: " &
myBooking.booking.booking_contact_id & " first area: " &
               myBooking.booking_areas(0))
    End If

Catch ex As Exception

    MsgBox("Exception: " + ex.Message)

End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl", array("trace"=>1,"exceptions"=>0));

// load bookingID 5 data
$myBookingStruct = $client->getBooking("username","password",5);

if($myBookingStruct instanceof SoapFault) {
    echo "SOAP Fault: ".$myBookingStruct->faultstring;
} elseif(isset($myBookingStruct->booking->booking_id)) {
    echo "Successfully received booking data";
} else {
  echo "Failed to get booking";
}
```

RETURNS

```
Object As BookingSOAP.bookingStruct
```

6.5 Method: getBookings

This function can be used to fetchmultiple booking records in their basic form. Results may be sorted by a chosen field and offset limit values provide for paging. The maximum limit value will be restricted to 25 records per query.

6.5.1 Example: getBookings

Function:`getBookings(username As String, password As String, booking_equal_arguments As BookingSOAP.bookingObject, booking_greater_equal_arguments As BookingSOAP.bookingObject, booking_less_equal_arguments As BookingSOAP.bookingObject,booking_in_sql_arguments As BookingSOAP.bookingObjectInSQL, result_options As BookingSOAP.resultOptions) As BookingSOAP.bookingStruct()`

.NET

```
Try

    Dim myService As New BookingSOAP.BookingManager
    Dim myBookings() As BookingSOAP.bookingStruct
    Dim myBookingEqualsArgs As New BookingSOAP.bookingObject
    Dim myBookingGreaterThanArgs As New BookingSOAP.bookingObject
    Dim myBookingLessThanArgs As New BookingSOAP.bookingObject
    Dim myBookingInSQLArgs As New BookingSOAP.bookingObjectInSQL
    Dim myResultOptions As New BookingSOAP.resultOptions

    ' restrict to unitID 67
    myBookingEqualsArgs.booking_unit_id = 67
    myBookingEqualsArgs.booking_unit_idSpecified = True

    ' set filter to be enquiries in  month of "March 2013"
    myBookingGreaterThanArgs.booking_enquiry_datetime = "2013-03-01 00:00:00"
    myBookingLessThanArgs.booking_enquiry_datetime = "2013-03-31 23:59:59"

    myBookings = myService.getBookings("username","password",
myBookingEqualsArgs,
myBookingGreaterThanArgs,
 myBookingLessThanArgs,
myBookingInSQLArgs, myResultOptions)

  If myBookings.Length < 1 Then
        MsgBox("Failed to get any bookings.")
    Else
        MsgBox("Success, found "&myBookings.Length &" records, "&
"First result booking_id: " &
myBookings(0).booking.booking_id & " with area count: " &
myBookings(0).booking_areas.Length)
    End If

Catch ex As Exception
    MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl",array("trace"=>1,"exceptions"=>0));

$myBookingEqualsArgs = array(); // exact match args

// restrict to unitIDs 67,68 and 69
$myBookingInSQLArgs['booking_unit_id'] = '67,68,69';

// set filter to be enquiries in  month of "March 2013"
$myBookingGreaterThanArgs['booking_enquiry_datetime'] = "2013-03-01 00:00:00";
$myBookingLessThanArgs['booking_enquiry_datetime'] = "2013-03-31 23:59:59";

// add some extra result option arguments
$myResultOptions['sort_by_element_name'] = 'booking_enquiry_datetime';
$myResultOptions['sort_direction'] = 'desc';
$myResultOptions['result_limit'] = '10';
$myResultOptions['result_offset'] = '0';

// run search
$myBookings = $client->getBookings("username","password",$myBookingEqualsArgs,
$myBookingGreaterThanArgs,
$myBookingLessThanArgs,
$myBookingInSQLArgs,
$myResultOptions);

if($myBookings instanceof SoapFault) {
    echo "SOAP Fault: ".$myBookings->faultstring;
} elseif(count($myBookings) >= '1') {
    echo "Success, found: ".count($myBookings);
} else {
    echo "Failed to find any bookings for given criteria";
}
```

RETURNS

```
Object As BookingSOAP.bookingStruct()
```

6.6 Method: getBookingsCount

This function can be used to return a total count of booking records available via a getBookings() call.

6.6.1 Example: getBookingsCount

Function:`getBookingsCount(username As String, password As String, booking_equal_arguments As BookingSOAP.bookingObject, booking_greater_equal_arguments As BookingSOAP.bookingObject, booking_less_equal_arguments As BookingSOAP.bookingObject,booking_in_sql_arguments As BookingSOAP.bookingObjectInSQL, result_options As BookingSOAP.resultOptions) count As Integer`

.NET

```
Try

    Dim myService As New BookingSOAP.BookingManager
    Dim myCount As Integer
    Dim myBookingEqualsArgs As New BookingSOAP.bookingObject
    Dim myBookingGreaterThanArgs As New BookingSOAP.bookingObject
    Dim myBookingLessThanArgs As New BookingSOAP.bookingObject
    Dim myBookingInSQLArgs As New BookingSOAP.bookingObjectInSQL
    Dim myResultOptions As New BookingSOAP.resultOptions

    ' restrict to unitID 67
    myBookingEqualsArgs.booking_unit_id = 67
    myBookingEqualsArgs.booking_unit_idSpecified = True

    ' set filter to be enquiries in  month of "March 2013"
    myBookingGreaterThanArgs.booking_enquiry_datetime = "2013-03-01 00:00:00"
    myBookingLessThanArgs.booking_enquiry_datetime = "2013-03-31 23:59:59"

    myCount = myService.getBookingsCount("username","password",
myBookingEqualsArgs,
myBookingGreaterThanArgs,
 myBookingLessThanArgs,
myBookingInSQLArgs, myResultOptions)

  If myCount >= 1 Then
        MsgBox("Found " & myCount & " bookings.")
    Else
        MsgBox("No bookings found.")
    End If

Catch ex As Exception
    MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl",array("trace"=>1,"exceptions"=>0));

$myBookingEqualsArgs = array(); // exact match args

// restrict to unitIDs 67,68 and 69
$myBookingInSQLArgs['booking_unit_id'] = '67,68,69';

// set filter to be enquiries in  month of "March 2013"
$myBookingGreaterThanArgs['booking_enquiry_datetime'] = "2013-03-01 00:00:00";
$myBookingLessThanArgs['booking_enquiry_datetime'] = "2013-03-31 23:59:59";

// add some extra result option arguments
$myResultOptions['sort_by_element_name'] = 'booking_enquiry_datetime';
$myResultOptions['sort_direction'] = 'desc';
$myResultOptions['result_limit'] = '10';
$myResultOptions['result_offset'] = '0';

// run search
$myCount = $client->getBookingsCount("username","password",$myBookingEqualsArgs,
$myBookingGreaterThanArgs,
$myBookingLessThanArgs, $myResultOptions);

if($myCount instanceof SoapFault) {
    echo "SOAP Fault: ".$myCount->faultstring;
} elseif($myCount >= '1') {
    echo "Success, found: ". $myCount;
} else {
    echo "Failed to find any bookings for given criteria";
}
```

RETURNS

```
Count As Integer
```

6.7 Method: getBookingNotes

This function can be used to fetch all notes for a single booking record. A booking note with a booking_note_pt_user_id equal to “0” are customer generated comments.

6.7.1 Example: getBookingNotes

Function:`getBookingNotes(username As String, password As String, booking_id As Integer) As BookingSOAP.bookingNoteObject()`

.NET

```
Try

    Dim myService As New BookingSOAP.BookingManager
    Dim myBookingNotes(0) As BookingSOAP.bookingNoteObject

    myBookingNotes = myService.getBookingNotes("username","password", 2)

    If myBookingNotes.Length >= 1 Then

        MsgBox("Success, 1st element booking_note_text: " &
myBookingNotes(0).booking_note_text)

    Else

        MsgBox("No booking notes for booking_id.")

    End If

Catch ex As Exception

    MsgBox("Exception: " + ex.Message)

End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl", array("trace"=>1,"exceptions"=>0));

// load bookingID 5's notes
$myBookingNotes = $client->getBookingNotes("username","password",5);

if($myBookingNotes instanceof SoapFault) {
    echo "SOAP Fault: ".$myBookingNotes->faultstring;
} elseif(count($myBookingNotes) >= 1) {
    echo "Successfully received booking data";
} else {
    echo "No booking notes for booking id";
}
```

RETURNS

```
Object As BookingSOAP.bookingNoteObject()
```

6.8 Method: getBookingTypes

This function can be used to fetch all available booking types and their associated IDs. This is for look up purposes only – if you require a new booking type please contact support@powertext.co.uk with your request.

6.8.1 Example: getBookingTypes

Function:`getBookingTypes(username As String, password As String) As BookingSOAP.bookingTypeObject()`

.NET

```
Try
    Dim myService As New BookingSOAP.BookingManager
    Dim myBookingTypes() As BookingSOAP.bookingTypeObject

    myBookingTypes = myService.getBookingTypes("username","password")

    If myBookingTypes.Length < 1 Then

        MsgBox("Failed to get booking types.")

    Else

        MsgBox("Success, first booking_type_id: " &
myBookingTypes(0).booking_type_id & " type name: " &
myBookingTypes(0).booking_type_name)

    End If

Catch ex As Exception

    MsgBox("Exception: " + ex.Message)

End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl", array("trace"=>1,"exceptions"=>0));

$myBookingTypes = $client->getBookingTypes("username","password");

if($myBookingTypes instanceof SoapFault) {
    echo "SOAP Fault: ".$myBookingTypes->faultstring;
} elseif(count($myBookingTypes) >= 1) {
    echo "Successfully received booking types";
} else {
    echo "No booking types??";
}
```

RETURNS

```
Object As BookingSOAP.bookingTypeObject()
```

6.9 Method: getBookingSources

This function can be used to fetch all available booking sources and their associated IDs. This is for look up purposes only – if you require a new booking source please contact support@powertext.co.uk with your request.

6.9.1 Example: getBookingSources

Function:`getBookingSources(username As String, password As String) As BookingSOAP.bookingSourceObject()`

.NET

```
Try
    Dim myService As New BookingSOAP.BookingManager
    Dim myBookingSources() As BookingSOAP.bookingSourceObject

    myBookingSources = myService.getBookingSources("username","password")

    If myBookingSources.Length < 1 Then

        MsgBox("Failed to get booking sources.")

    Else

        MsgBox("Success, first booking_source_id: " &
 myBookingSources(0).booking_enquiry_source_id & " source name: "
& myBookingSources(0).booking_enquiry_source_name)

    End If

Catch ex As Exception

    MsgBox("Exception: " + ex.Message)

End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl", array("trace"=>1,"exceptions"=>0));

$myBookingSources = $client->getBookingSources("username","password");

if($myBookingSources instanceof SoapFault) {
    echo "SOAP Fault: ".$myBookingSources->faultstring;
} elseif(count($myBookingSources) >= 1) {
    echo "Successfully received booking sources";
} else {
    echo "No booking sources?";
}
```

RETURNS

```
Object As BookingSOAP.bookingSourceObject()
```

6.10 Method: createBookingEnquiry

This function can be used to add a new "booking enquiry" record, it will return the provided bookingID on success. Before calling the function we must have a valid contact_id – if this is a new contact the createContact must be called first to obtain a new contact_id.

A "booking enquiry" is the same as a booking except "party_size" and "booking_party_datetime" are not required. To update booking enquiries and create notes for "booking enquiries" you may still use other booking API calls.

6.10.1 Example: createBookingEnquiry

Function:`createBookingEnquiry(username As String, password As String, booking As BookingSOAP.bookingEnquiryObject, booking_areas() As Integer) booking_id As Integer`

.NET

```
Try

    Dim myService As New BookingSOAP.BookingManager
    Dim myBooking As New BookingSOAP.bookingEnquiryObject
    Dim myBookingAreas(1) As Integer
    Dim myBookingID As Integer

    myBookingAreas(0) = "101" ' reserve area 101 and 102"
    myBookingAreas(1) = "102" ' reserve area 101 and 102

    myBooking.booking_contact_id = "5282391"
    myBooking.booking_contact_idSpecified = True
    myBooking.booking_enquiry_source_id = "2"
    myBooking.booking_enquiry_source_idSpecified = True
    myBooking.booking_stage_value = "1"
    myBooking.booking_stage_valueSpecified = True
    myBooking.booking_type_id = "2"
    myBooking.booking_type_idSpecified = True
    myBooking.booking_unit_id = "67"
    myBooking.booking_unit_idSpecified = True
    myBooking.booking_deposit_paid = 0
    myBooking.booking_deposit_paidSpecified = True
    myBooking.booking_hpbr_drink = 0
    myBooking.booking_hpbr_drinkSpecified = True
    myBooking.booking_hpbr_food = 0
    myBooking.booking_hpbr_foodSpecified = True
    myBooking.booking_hpbr_entertainment = 0
    myBooking.booking_hpbr_entertainmentSpecified = True

    myBooking.booking_enquiry_datetime = "2013-03-21 21:00:06"
    myBooking.booking_data_element_03 = "beer"
    myBooking.booking_data_element_20 = "steak"

    myBookingID = myService.createBookingEnquiry("username", "password",
                                                 myBooking, myBookingAreas)
    If myBookingID >= 1 Then
        MsgBox("Success, booking_id: " & myBookingID)
    Else
        MsgBox("Failed to create booking.")
    End If

Catch ex As Exception
    MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$client = new SoapClient("../Bookings.wsdl",array("trace"=>1,"exceptions"=>0));

$booking = array(
   'booking_contact_id'=>'5282391',
   'booking_enquiry_source_id'=>'2',
   'booking_stage_value'=> '1',
   'booking_type_id'=> '2',
   'booking_unit_id'=> '67',
   'booking_enquiry_datetime'=> "2013-03-21 21:00:06",
   'booking_deposit_paid'=> '0',
   'booking_hpbr_drink'=> '0',
   'booking_hpbr_food'=> '0',
   'booking_hpbr_entertainment'=> '0',
   'booking_data_element_03'=> "wine",
   'booking_data_element_20'=> "chicken" );

$booking_areas = array('108','109');

$myBookingID = $client->createBookingEnquiry("username","password",
                                       $booking,$booking_areas);

if($myBookingID instanceof SoapFault || $myBookingID < '1') {
    echo "SOAP Fault: ".$myBookingID->faultstring;
} else {
    echo "Success, new booking_id: ".$myBookingID;
}
```

RETURNS

```
booking_id As Integer
```

7. Search API

The search api allows basic searches to be performed on contact records.

File:https://secure.powertext.co.uk/SOAP/V2/Search.wsdl**Namespace:** SearchSOAP

7.1 Method: searchContactEmail

This function can be used to obtain an existing contact record, you must however provide the full email address of the contact in the call.

7.1.1 Example: searchContactEmail

Function:`searchContactEmail (username As String, password As String, email As String) As SearchSOAP.contactStruct`

.NET

```
        Try

            Dim myService As New SearchSOAP.SearchManager
            Dim myContact As New SearchSOAP.contactStruct

            myContact = myService.searchContactEmail("username", "password", 
"find@example.com")

            If myContact Is Nothing Then
                MsgBox("searchContact found no contact.")
            Else
                MsgBox("searchContact Success, element email: " & 
                                   myContact.email)
            End If

        Catch ex As Exception

            MsgBox("searchContactExample() Exception: " + ex.Message)

        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
        $client = new SoapClient($wsdl);


$myContact = $client->searchContactEmail("username","password",
"find@example.com" );

if(!isset($myContact->contactid)) {
          echo "Failed: ".$myContact;
} else {
echo "Success, element email: ".$myContact->email;
}
```

RETURNS

```
object->contactid As Integer
object->uid As String
object->title As String
object->firstname As String
object->lastname As String
object->gender As String
object->dob As String
object->mobilenumber As String
object->landnumber As String
object->email As String
object->buildingname As String
object->buildingnumstreet As String
object->city As String
object->locality As String
object->county As String
object->postcode As String
object->country As String
object->membershiptype As String
object->membershipnumber As String
object->allowsms As String
object->allowemail As String
object->allowsnailmail As String
object->allowcall As String
object->sourceid As String
object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
object->groups As Array of Integers
```

7.2 Method: searchContactMobile

This function can be used to obtain an existing contact record, you must however provide the full mobile telephone number of the contact in the call.

7.2.1 Example: searchContactMobile

Function:`searchContactMobile (username As String, password As String, mobilenumber As String) As SearchSOAP.contactStruct`

.NET

```
        Try

            Dim myService As New SearchSOAP.SearchManager
            Dim myContact As New SearchSOAP.contactStruct

            myContact = myService.searchContactMobile("username", "password",
"4477111111")

            If myContact Is Nothing Then
                MsgBox("searchContact found no contact.")
            Else
                MsgBox("searchContact Success, element email: " &
                                   myContact.email)
            End If

        Catch ex As Exception

            MsgBox("searchContactExample() Exception: " + ex.Message)

        End Try
```

.PHP

```
        $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
        $client = new SoapClient($wsdl);


$myContact = $client->searchContactEmail("username","password",
"4477111111");

if(!isset($myContact->contactid)) {
          echo "Failed: ".$myContact;
} else {
echo "Success, element email: ".$myContact->email;
}
```

RETURNS

```
object->contactid As Integer
object->uid As String
object->title As String
object->firstname As String
object->lastname As String
object->gender As String
object->dob As String
object->mobilenumber As String
object->email As String
object->buildingname As String
object->buildingnumstreet As String
object->city As String
object->locality As String
object->county As String
object->postcode As String
object->country As String
object->membershiptype As String
object->membershipnumber As String
object->allowsms As String
object->allowemail As String
object->allowsnailmail As String
object->allowcall As String
object->sourceid As String
object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
object->groups As Array of Integers
```

7.3 Method: searchContactName

This function can be used to find existing contact records, you must however provide a searcharble string in either firstname, lastname or both. If you only provide a firstname or lastname you must provide the other as a blank string ie. ""

7.3.1 Example: searchContactName

Function:searchContactName (username As String, password As String, fistname As String, lastname As string, result_options As SearchSOAP.resultOptions) As SearchSOAP.contactStruct()

.NET

```
    Try
        Dim myService As New SearchSOAP.SearchManager
        Dim myResponse(0) As SearchSOAP.ContactData
        Dim myResultOptions As New SearchSOAP.resultOptions

        myResultOptions.result_offset = "0"
        myResultOptions.result_offsetSpecified = True

        myResultOptions.result_limit = "10"
        myResultOptions.result_limitSpecified = True

        myResultOptions.sort_by_element_name = "Firstname"
        myResultOptions.sort_direction = "desc"

        myContacts = myService.searchContactName("username", "password",
                                                 "", "Smith",
                                                 myResultOptions)

If myContacts.Length >= 1 Then
            MsgBox("first row contactid: " & myResponse(0).contactid)
        Else
            MsgBox("nothing found")
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
    $client = new SoapClient($wsdl);
    // add some extra result option arguments
    $myResultOptions['sort_by_element_name'] = 'Firstname';
    $myResultOptions['sort_direction'] = 'desc';
    $myResultOptions['result_limit'] = '10';
    $myResultOptions['result_offset'] = '0';

    $myResponse = $client->searchContactName("username", "password",
                                              "john","smith", $myResultOptions);
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
  object->contactid As Integer
  object->uid As String
  object->title As String
  object->firstname As String
  object->lastname As String
  object->gender As String
  object->dob As String
  object->mobilenumber As String
  object->landnumber As String
  object->email As String
  object->buildingname As String
  object->buildingnumstreet As String
  object->city As String
  object->locality As String
  object->county As String
  object->postcode As String
  object->country As String
  object->membershiptype As String
  object->membershipnumber As String
  object->allowsms As String
  object->allowemail As String
  object->allowsnailmail As String
  object->allowcall As String
  object->sourceid As String
  object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
  object->groups As Array of Integers
)
```

7.4 Method: searchContactDOB

This function can be used to find existing contact records based on their date of birth. you must provide an operator and a date string. Operators can be one of the following: = (equals) > (greater than) < (less than)

7.4.1 Example: searchContactDOB

Function:`searchContactDOB (username As String, password As String, dob As String, operator As string, result_options As SearchSOAP.resultOptions) As SearchSOAP.contactStruct()`

.NET

```
    Try
        Dim myService As New SearchSOAP.SearchManager
        Dim myResponse(0) As SearchSOAP.ContactData
        Dim myResultOptions As New SearchSOAP.resultOptions

        myResultOptions.result_offset = "0"
        myResultOptions.result_offsetSpecified = True

        myResultOptions.result_limit = "20"
        myResultOptions.result_limitSpecified = True

        myResultOptions.sort_by_element_name = "DOB"
        myResultOptions.sort_direction = "asc"

        myContacts = myService.searchContactDOB("username", "password",
                                                "1992-12-22",">=",
                                                myResultOptions)

        If myContacts.Length >= 1 Then
            MsgBox("first row contactid: " & myResponse(0).contactid)
        Else
            MsgBox("nothing found")
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
    $client = new SoapClient($wsdl);
    // add some extra result option arguments
    $myResultOptions['sort_by_element_name'] = 'DOB';
    $myResultOptions['sort_direction'] = 'asc';
    $myResultOptions['result_limit'] = '20';
    $myResultOptions['result_offset'] = '0';
    $myResponse = $client->searchContactDOB("username", "password",
                                            "1992-12-22",">=",$myResultOptions);
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
  object->contactid As Integer
  object->uid As String
  object->title As String
  object->firstname As String
  object->lastname As String
  object->gender As String
  object->dob As String
  object->mobilenumber As String
  object->landnumber As String
  object->email As String
  object->buildingname As String
  object->buildingnumstreet As String
  object->city As String
  object->locality As String
  object->county As String
  object->postcode As String
  object->country As String
  object->membershiptype As String
  object->membershipnumber As String
  object->allowsms As String
  object->allowemail As String
  object->allowsnailmail As String
  object->allowcall As String
  object->sourceid As String
  object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
  object->groups As Array of Integers
)
```

7.5 Method: searchContactUDF

This function can be used to find existing contact records with a UDF field matching the specified value. It is advised that this call only be made on UDF fields which are set as Unique. Other UDF fields may take longer to return results.

7.5.1 Example: searchContactUDF

Function:`searchContactUDF (username As String, password As String, udfnameid As Integer, udfvalue As String, result_options As SearchSOAP.resultOptions) As SearchSOAP.contactStruct()`

.NET

```
    Try
        Dim myService As New SearchSOAP.SearchManager
        Dim myResponse(0) As SearchSOAP.ContactData
        Dim myResultOptions As New SearchSOAP.resultOptions

        myResultOptions.result_offset = "0"
        myResultOptions.result_offsetSpecified = True

        myResultOptions.result_limit = "50"
        myResultOptions.result_limitSpecified = True

        myResultOptions.sort_by_element_name = "ContactID"
        myResultOptions.sort_direction = "asc"

        myContacts = myService.searchContactUDF("username", "password",
                                                "1234","member",
                                                myResultOptions)

        If myContacts.Length >= 1 Then
            MsgBox("first row contactid: " & myResponse(0).contactid)
        Else
            MsgBox("nothing found")
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
    $client = new SoapClient($wsdl);
    // add some extra result option arguments
    $myResultOptions['sort_by_element_name'] = 'ContactID';
    $myResultOptions['sort_direction'] = 'asc';
    $myResultOptions['result_limit'] = '50';
    $myResultOptions['result_offset'] = '0';
    $myResponse = $client->searchContactUDF("username", "password",
                                             "1234","member",$myResultOptions);
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
  object->contactid As Integer
  object->uid As String
  object->title As String
  object->firstname As String
  object->lastname As String
  object->gender As String
  object->dob As String
  object->mobilenumber As String
  object->landnumber As String
  object->email As String
  object->buildingname As String
  object->buildingnumstreet As String
  object->city As String
  object->locality As String
  object->county As String
  object->postcode As String
  object->country As String
  object->membershiptype As String
  object->membershipnumber As String
  object->allowsms As String
  object->allowemail As String
  object->allowsnailmail As String
  object->allowcall As String
  object->sourceid As String
  object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
  object->groups As Array of Integers
)
```

7.6 Method: searchContactUpdated

This function can be used to find existing contact records that have been updated since the given "timestamp" value.

7.6.1 Example: searchContactUpdated

Function:`searchContactUpdated (username As String, password As String, timestamp As Integer, result_options As SearchSOAP.resultOptions) As SearchSOAP.contactStruct()`

.NET

```
    Try
        Dim myService As New SearchSOAP.SearchManager
        Dim myResponse(0) As SearchSOAP.ContactData
        Dim myResultOptions As New SearchSOAP.resultOptions

        myResultOptions.result_offset = "0"
        myResultOptions.result_offsetSpecified = True

        myResultOptions.result_limit = "50"
        myResultOptions.result_limitSpecified = True

        myResultOptions.sort_by_element_name = "ContactID"
        myResultOptions.sort_direction = "asc"

        myResponse = myService.searchContactUpdated("username", "password",
                                                     "1340695138",
                                                     myResultOptions)
        If myContacts.Length >= 1 Then
            MsgBox("first row contactid: " & myResponse(0).contactid)
        Else
            MsgBox("nothing found")
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
    $client = new SoapClient($wsdl);
    // add some extra result option arguments
    $myResultOptions['sort_by_element_name'] = 'ContactID';
    $myResultOptions['sort_direction'] = 'desc';
    $myResultOptions['result_limit'] = '10';
    $myResultOptions['result_offset'] = '0';
    $myResponse = $client->searchContactUpdated("username", "password",
                                            "1340695138",$myResultOptions);
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
  object->contactid As Integer
  object->uid As String
  object->title As String
  object->firstname As String
  object->lastname As String
  object->gender As String
  object->dob As String
  object->mobilenumber As String
  object->landnumber As String
  object->email As String
  object->buildingname As String
  object->buildingnumstreet As String
  object->city As String
  object->locality As String
  object->county As String
  object->postcode As String
  object->country As String
  object->membershiptype As String
  object->membershipnumber As String
  object->allowsms As String
  object->allowemail As String
  object->allowsnailmail As String
  object->allowcall As String
  object->sourceid As String
  object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
  object->groups As Array of Integers
)
```

7.7 Method: searchContactAdded

This function can be used to find existing contact records that have been added since the given "timestamp" value.

7.7.1 Example: searchContactAdded

Function:`searchContactAdded (username As String, password As String, timestamp As Integer, result_options As SearchSOAP.resultOptions) As SearchSOAP.contactStruct()`

.NET

```
    Try
        Dim myService As New SearchSOAP.SearchManager
        Dim myResponse(0) As SearchSOAP.ContactData
        Dim myResultOptions As New SearchSOAP.resultOptions

        myResultOptions.result_offset = "0"
        myResultOptions.result_offsetSpecified = True

        myResultOptions.result_limit = "50"
        myResultOptions.result_limitSpecified = True

        myResultOptions.sort_by_element_name = "ContactID"
        myResultOptions.sort_direction = "asc"

        myResponse = myService.searchContactAdded("username", "password",
                                                  "1340695138",
                                                  myResultOptions)
        If myContacts.Length >= 1 Then
            MsgBox("first row contactid: " & myResponse(0).contactid)
        Else
            MsgBox("nothing found")
        End If

    Catch ex As Exception
        MsgBox("Exception: " + ex.Message)
    End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
    $client = new SoapClient($wsdl);
    // add some extra result option arguments
    $myResultOptions['sort_by_element_name'] = 'ContactID';
    $myResultOptions['sort_direction'] = 'desc';
    $myResultOptions['result_limit'] = '10';
    $myResultOptions['result_offset'] = '0';
    $myResponse = $client->searchContactAdded("username", "password",
                                            "1340695138",$myResultOptions);
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
  object->contactid As Integer
  object->uid As String
  object->title As String
  object->firstname As String
  object->lastname As String
  object->gender As String
  object->dob As String
  object->mobilenumber As String
  object->landnumber As String
  object->email As String
  object->buildingname As String
  object->buildingnumstreet As String
  object->city As String
  object->locality As String
  object->county As String
  object->postcode As String
  object->country As String
  object->membershiptype As String
  object->membershipnumber As String
  object->allowsms As String
  object->allowemail As String
  object->allowsnailmail As String
  object->allowcall As String
  object->sourceid As String
  object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
  object->groups As Array of Integers
)
```

7.8 Method: searchContactGroups

This function can be used to find existing contact records that are assigned to a specific groupID or set of groupIDs.

7.8.1 Example: searchContactGroups

Function:`searchContactGroups (username As String, password As String, group_ids As Array of Integer, result_options As SearchSOAP.resultOptions) As SearchSOAP.contactStruct()`

.NET

```
Try

    Dim myService As New SearchSOAP.SearchManager
    Dim myContacts() As SearchSOAP.contactObject
    Dim myResultOptions As New SearchSOAP.resultOptions
    Dim groupIDs(1) As Integer
    groupIDs(0) = 275
    groupIDs(1) = 277

    myResultOptions.result_offset = "0"
    myResultOptions.result_offsetSpecified = True
    myResultOptions.result_limit = "10"
    myResultOptions.result_limitSpecified = True
    myResultOptions.sort_by_element_name = "Firstname"
    myResultOptions.sort_direction = "desc"

    myContacts = myService.searchContactGroups("username", "password",
                                          groupIDs, myResultOptions)
    If myContacts.Length >= 1 Then
        MsgBox("searchContact Success, element email: " &
                           myContacts(0).email)
    Else
        MsgBox("no contacts found")
    End If

Catch ex As Exception

    MsgBox("exception: " + ex.Message)

End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
    $client = new SoapClient($wsdl);
    // add some extra result option arguments
    $myResultOptions['sort_by_element_name'] = 'ContactID';
    $myResultOptions['sort_direction'] = 'desc';
    $myResultOptions['result_limit'] = '10';
    $myResultOptions['result_offset'] = '0';
    $myResponse = $client->searchContactGroups("username", "password",
                                              array(275,277),$myResultOptions);
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
  object->contactid As Integer
  object->uid As String
  object->title As String
  object->firstname As String
  object->lastname As String
  object->gender As String
  object->dob As String
  object->mobilenumber As String
  object->landnumber As String
  object->email As String
  object->buildingname As String
  object->buildingnumstreet As String
  object->city As String
  object->locality As String
  object->county As String
  object->postcode As String
  object->country As String
  object->membershiptype As String
  object->membershipnumber As String
  object->allowsms As String
  object->allowemail As String
  object->allowsnailmail As String
  object->allowcall As String
  object->sourceid As String
  object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
  object->groups As Array of Integers
)
```

7.9 Method: searchContactPostCode

This function can be used to find existing contact records that match the given postcode. Postcode can be passed in as either the first part (region) or as a full postcode.

7.9.1 Example: searchContactGroups

Function:`searchContactGroups (username As String, password As String, postcode As String, result_options As SearchSOAP.resultOptions) As SearchSOAP.contactStruct()`

.NET

```
Try

    Dim myService As New SearchSOAP.SearchManager
    Dim myContacts() As SearchSOAP.contactObject
    Dim myResultOptions As New SearchSOAP.resultOptions
    Dim postcode As String
    postcode = "S11 7EB"

    myResultOptions.result_offset = "0"
    myResultOptions.result_offsetSpecified = True
    myResultOptions.result_limit = "10"
    myResultOptions.result_limitSpecified = True
    myResultOptions.sort_by_element_name = "Firstname"
    myResultOptions.sort_direction = "desc"

    myContacts = myService.searchContactPostCode("username", "password",
                                          postcode, myResultOptions)
    If myContacts.Length >= 1 Then
        MsgBox("searchContact Success, element email: " &
                           myContacts(0).email)
    Else
        MsgBox("no contacts found")
    End If

Catch ex As Exception

    MsgBox("exception: " + ex.Message)

End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
    $client = new SoapClient($wsdl);
    // add some extra result option arguments
    $myResultOptions['sort_by_element_name'] = 'ContactID';
    $myResultOptions['sort_direction'] = 'desc';
    $myResultOptions['result_limit'] = '10';
    $myResultOptions['result_offset'] = '0';
    $myResponse = $client->searchContactPostCode("username", "password",
                                              "S11 7EB",$myResultOptions);
    if($myResponse instanceof SoapFault) {
        die("Exception: ".$myResponse->getMessage());
    } elseif(is_array($myResponse)) {
        echo "<pre>response: ".print_r($myResponse,1)."</pre>";
    } else {
        echo "bad response: ".$myResponse;
    }
```

RETURNS

```
Array of Object (
  object->contactid As Integer
  object->uid As String
  object->title As String
  object->firstname As String
  object->lastname As String
  object->gender As String
  object->dob As String
  object->mobilenumber As String
  object->landnumber As String
  object->email As String
  object->buildingname As String
  object->buildingnumstreet As String
  object->city As String
  object->locality As String
  object->county As String
  object->postcode As String
  object->country As String
  object->membershiptype As String
  object->membershipnumber As String
  object->allowsms As String
  object->allowemail As String
  object->allowsnailmail As String
  object->allowcall As String
  object->sourceid As String
  object->udfs As Array of Objects (o->udfnameid, o->data, o->type)
  object->groups As Array of Integers
)
```

7.10 Method: searchContactGroupsCount

This function can be used to count existing contact records that are assigned to a specific groupID or set of groupIDs.

7.10.1 Example: searchContactGroupsCount

Function:`searchContactGroupsCount (username As String, password As String, group_ids As Array of Integer) As Integer`

.NET

```
Try

    Dim myService As New SearchSOAP.SearchManager
    Dim myCount As Integer
    Dim groupIDs(1) As Integer
    groupIDs(0) = 275
    groupIDs(1) = 277

    myCount = myService.searchContactGroupsCount("username", "password",
                                                 groupIDs)
    If myCount >= 1 Then
        MsgBox("found " & myCount & "contacts")
    Else
    MsgBox("no contacts found")
    End If

Catch ex As Exception

    MsgBox("exception: " + ex.Message)

End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/Search.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->searchContactGroupsCount("username", "password",
array(275,277));
if($myResponse instanceof SoapFault) {
    die("exception: ".$myResponse->getMessage());
} elseif($myResponse >= 1) {
    echo "<pre>found: ". $myResponse." records</pre>";
} else {
    echo "<pre>no records found</pre>";
}
```

RETURNS

```
Count As Integer
```

8. Export Call List API

The export call list api allows for retrieval of, “phone call” broadcast type, contact data sets. It also provides a mechanism for acknowledging reciept of contact data set chunks.

File:https://secure.PowerText.co.uk/SOAP/V2/CallList.wsdl**Namespace:**callListSOAP

8.1 Method: getCallLists

This function is used to fetch a list of currently unprocessed and part processed data sets, it will provide a unique result set ID for each set of data currently awaiting export, along with the number of processed and unprocessed contacts along with the date and time of broadcast.

8.1.1 Example: getCallLists

Function:`getCallLists(username As String, password As String) As callListSOAP.callListObject()`

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/CallList.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->getCallLists("username","password");

if($myResponse instanceof SoapFault) {
    echo "SOAP Fault: ".$myResponse->faultstring;
} elseif(count($myResponse) >= '1') {
    echo "Success, found: ".count($myResponse);
} else {
    echo "Failed to find any call lists";
}
```

RETURNS

```
Array of Object (
            object->call_list_id  As Integer
            object->call_list_broadcast_id  As Integer
            object->call_list_unit_id  As Integer
            object->call_list_name As string
            object->call_list_ref As String
            object->call_list_created As String
            object->call_list_queued_calls As Integer
            object->call_list_dequeued_calls As Integer
            object->call_list_total_calls As Integer
)
```

8.2 Method: getCallListItems

This function can be used to retrieve a block of contact data from a data set. The block size is hard limited to 250 contact records per call. Multiple calls must be made in order to extract all contact records from a large "phone call" broadcast data set.

8.2.1 Example: getCallListItems

Function:`getCallListItems (username As String, password As String, call_list_id as Integer, offset, limit) As callListSOAP.contactCallObject()`

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/CallList.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->getCallListItems("username","password",1,0,100);
if($myResponse instanceof SoapFault) {
    echo "SOAP Fault: ".$myResponse->faultstring;
} elseif(count($myResponse) >= '1') {
    echo "Success, found: ".count($myResponse)
} else {
    echo "Failed to find any call items in list for given offset / limit";
}
```

RETURNS

```
Array of Object (
 object->list_item_id As Integer
 object->list_item_ack As String
 object->list_item_contact_id As Integer
 object->list_item_unit_id  As Integer
 object->firstname As String
 object->lastname As String
 object->gender As String
 object->dob As String
 object->mobilenumber As String
 object->allowcall As String
)
```

8.3 Method: ackCallListItems

This function is used to acknowledge receipt of a block of contact data from a data set. This must be called once you have received a block of data from a call to getCallListItems ()

8.3.1 Example: ackCallListItems

Function:`ackCallListItems (username As String, password As String, call_list_id as Integer, offset, limit) As Integer`

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/CallList.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->ackCallListItems("username","password",1,0,100);
if($myResponse instanceof SoapFault) {
    echo "SOAP Fault: ".$myResponse->faultstring;
} elseif($myResponse == '100') {
    echo "Success, acknowledged records.";
} else {
    echo "Failed to acknowledge any items for given offset / limit";
}
```

RETURNS

```
"100" As Integer  (on success)
```

9. Unique Codes API

The unique codes api allows for retrieval of unique code groups, assigned “unique codes” and associated contact data sets. It also provides a mechanism for searching and redeeming unique codes.

File:https://secure.PowerText.co.uk/SOAP/V2/ UniqueCodes.wsdl**Namespace:**UniqueCodeSOAP

9.1 Method: accountGetUniqueCodeGroups

This function is used to fetch a list of non expired code groups.It will return a unique ucGID for each code group along with other code group specific fields.

9.1.1 Example: accountGetUniqueCodeGroups

Function:`accountGetUniqueCodeGroups (username As String, password As String) As UniqueCodeSOAP.uniqueCodeGroupObject()`

.NET

```
Dim myService As New UniqueCodeSOAP.UniqueCodes
Dim myCodeGroups() As UniqueCodeSOAP.uniqueCodeGroupObject
myCodeGroups = myService.accountGetUniqueCodeGroups("username", "password")
If myCodeGroups.Length >= 1 Then
    MsgBox("found " & myCodeGroups.Length)
End If
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UniqueCodes.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->accountGetUniqueCodeGroups("username","password");
if($myResponse instanceof SoapFault) {
die( "SOAP Fault: ".$myResponse->faultstring );
} elseif(count($myResponse) >= '1') {
    echo "Success, found: ".count($myResponse);
}
```

RETURNS

```
Array of Object (
    object->ucgid As Integer
    object->groupid As Integer
    object->customerdescription As String
    object->expirydate As String
    object->eflyerreplyid As Integer
    object->smsreplyid As Initeger
)
```

9.2 Method: uniqueCodeGetContact

This function is used to find the contactID assigned the given unique code within the given unique code group, a value of 0 will be returned where the unique code can not be found within the unique code group or it is not currently assigned to a contact.

9.2.1 Example: uniqueCodeGetContact

Function:`uniqueCodeGetContact(username As String, password As String, uniquecode As String, ucgid As Integer) As Integer`

.NET

```
Dim myService As New UniqueCodeSOAP.UniqueCodes
Dim myContactID As Integer
myContactID = myService.uniqueCodeGetContact("username", "password",
"LVN40051 - NGQKFDED", 5)
If myContactID > 0 Then
    MsgBox("assigned to " & myContactID)
End If
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UniqueCodes.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->uniqueCodeGetContact("username","password",
"LVN40051 - NGQKFDED",5)
if($myResponse instanceof SoapFault) {
die( "SOAP Fault: ".$myResponse->faultstring );
} elseif($myResponse >= '1') {
    echo "Success, found: ".$myResponse;
}
```

RETURNS

```
ContactID As Integer
```

9.3 Method: contactGetUniqueCodes

This function is used to obtain a full list of unique codes and associated data elements, which have been assigned to a given contact.

9.3.1 Example: contactGetUniqueCodes

Function:`contactGetUniqueCodes(username As String, password As String, contactid As Integer) As UniqueCodeSOAP.uniqueCodeContactObject()`

.NET

```
Dim myService As New UniqueCodeSOAP.UniqueCodes
Dim myContactCodes() As UniqueCodeSOAP.uniqueCodeContactObject
myContactCodes = myService.contactGetUniqueCodes("username", "passwd", 3742154)
If myContactCodes.Length >= 1 Then
    MsgBox("found " & myContactCodes.Length)
Else
    MsgBox("not found")
End If
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UniqueCodes.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->contactGetUniqueCodes("username","password", 3742154);
if($myResponse instanceof SoapFault) {
die( "SOAP Fault: ".$myResponse->faultstring );
} elseif(count($myResponse) >= '1') {
    echo "Success, found: ".count($myResponse);
}
```

RETURNS

```
Array of Object (
    object->ucgid As Integer
    object->uniquecode As String
    object->contactid As Integer
    object->assigneddate As String
    object->redeemeddate As String
    object->redeemedatunitid As Integer
    object->customerdescription As String
    object->expirydate As String
)
```

9.4 Method: uniqueCodeGetData

This function is used to find relevant data for a given unique code within a unique code group.

9.4.1 Example: uniqueCodeGetData

Function:`uniqueCodeGetData(username As String, password As String, uniquecode As String, ucgid As Integer) As UniqueCodeSOAP.uniqueCodeContactObject`

.NET

```
Dim myService As New UniqueCodeSOAP.UniqueCodes
Dim myContact As UniqueCodeSOAP.uniqueCodeObject
myContact = myService.uniqueCodeGetData("username", "password",
                                        "LVN40051 - NGQKFDED", 5)
If myContact.contactid >= 1 Then
    MsgBox("found " & myContact.contactid)
Else
    MsgBox("not found")
End If
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UniqueCodes.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->uniqueCodeGetData("username","password",
"LVN40051 - NGQKFDED",5);
if($myResponse instanceof SoapFault) {
die( "SOAP Fault: ".$myResponse->faultstring );
} elseif(isset($myResponse->ucgid)) {
    echo "Success, found: ".print_r($myResponse,1);
}
```

RETURNS

```
object->ucgid As Integer
object->uniquecode As String
object->contactid As Integer
object->assigneddate As String
object->redeemeddate As String
object->redeemedatunitid As Integer
object->customerdescription As String
object->expirydate As String
```

9.5 Method: redeemUniqueCode

This function is used to find relevant data for a given unique code within a unique code group.

9.5.1 Example: redeemUniqueCode

Function:`redeemUniqueCode(username As String, password As String, uniquecode As String, ucgid As Integer, timestamp As Integer, unitid As Integer) As Integer`

.NET

```
Dim myService As New UniqueCodeSOAP.UniqueCodes
Dim myResult As Integer
Dim currentTimestamp As Integer
currentTimestamp = DateDiff("s", "01/01/1970 00:00:00", Now())

Try
    myResult = myService.redeemUniqueCode("username", "password",
                                          "LVN44447 - RRFKNDRS", 9,
currentTimestamp, 986)
    If myResult.Equals(100) Then
        MsgBox("Success")
    Else
        MsgBox("Failure")
    End If
Catch ex As Exception
    MsgBox("Failure: " + ex.Message) ' most likely that already redeemed.
End Try
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UniqueCodes.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client-> redeemUniqueCode ("username","password",
"LVN40051 - NGQKFDED", 9,
                                          mktime(), 986);
if($myResponse instanceof SoapFault) {
die( "SOAP Fault: ".$myResponse->faultstring );
} elseif($myResponse == 100)) {
    echo "Success";
}
```

RETURNS

```
"100" on Success
```

9.6 Method: uniqueCodeGetAssigned

This function is used to find all codes from within a code group which have been assigned to a contactID, the function takes an offset and limit parameter, the maximum records returned per call is 500. To retrieve all records you must step through the results in blocks using the offset, limit parameters provided.

9.6.1 Example: uniqueCodeGetAssigned

Function: `uniqueCodeGetAssigned(username As String, password As String, ucgid As Integer, offset As Integer, limit As Integer) As UniqueCodeSOAP.uniqueCodeObject()`

.NET

```
Dim myService As New UniqueCodeSOAP.UniqueCodes
Dim myContacts() As UniqueCodeSOAP.uniqueCodeObject
myContacts = myService.uniqueCodeGetAssigned("username", "Password",
                                             9, 0, 100)
If myContacts.Length >= 1 Then
    MsgBox("uniqueContactGetAssigned, found " & myContacts.Length)
Else
    MsgBox("uniqueContactGetAssigned, not found")
End If
```

.PHP

```
$wsdl = "https://secure.powertext.co.uk/SOAP/V2/UniqueCodes.wsdl";
$client = new SoapClient($wsdl);
$myResponse = $client->uniqueCodeGetAssigned("username","password",
   9, 0, 100);
if($myResponse instanceof SoapFault) {
die( "SOAP Fault: ".$myResponse->faultstring );
} elseif(count($myResponse)>= 1)) {
    echo "Found: ".print_r($myResponse,1);
}
```

RETURNS

```
Array of Object (
    object->ucgid As Integer
    object->uniquecode As String
    object->contactid As Integer
    object->assigneddate As String
    object->redeemeddate As String
    object->redeemedatunitid As Integer
)
```

10. Feedback API

The Feedback API provides methods for saving feedback ratings and comments, and searching for feedback by feedback instance or contact.

10.1 Setting up Feedback

The following section explains the data required to set up a feedback form

10.1.1 Feedback Types

A “feedback type” can be any one of the following values, “positive”, “negative”, “neutral”, “unknown”. This is an overall view of the feedback based on your own external rules.

10.1.2 Feedback Source

A “feedback source” is used to indicate the original source of the feedback. All feedback sources and their IDs are preset within PowerText, these can be obtained via the getFeedbackSources() API. If none of the “feedback sources” suit your requirement(s), you may contact support@powertext.co.uk and request the addition of a new “feedback source”.

10.1.3 Feedback Category / Ratings

“Feedback Ratings” and “Feedback Categories” are used to store the feedback rating on a per category basis. For instance you may ask for feedback ratings on several categories such as food, drink, service and each category will require it’s own feedback rating score.

All “feedback categories” and their IDs are preset within PowerText, these can be obtained via the getFeedbackCategories() API. If none of the “feedback categories” suit your requirement(s), you may contact support@powertext.co.uk and request the addition of a new “feedback category”.

10.1.4 Feedback Comments

A feedback record can have multiple comments attached. Each feedback comment will contain a textual identifier to help keep track of who made the comment. Feedback comments can be obtained via the getFeedbackComments() API

10.1.5 Example Feedback Questionnaire

Feedback form on your website:

Question description	**Question type (rating/comment)**	**Feedback Source ID**	**Feedback Type ID**	**Feedback Category**	**Feedback Ratings**	**Feedback Comment**
Were you satisfied with your reception at our restaurant?	Rating	1 - Website	4 - unknown	New OR 5 - Welcome	Very Satisfied / Satisfied / No Opinion / Not Satisfied/ Very Unsatisfied	
How could we improve you welcome	Comment	1 - Website	4 - unknown	New OR 5 - Welcome		Free text - Have the reception near the bar area
How did you enjoy your meal? Add comments if necessary.	Rating	1 - Website	4 - unknown	New OR 1 - Food	Very Satisfied / Satisfied / No Opinion / Not Satisfied/ Very Unsatisfied	Starters were a little on the small side.
Was the drinks choice available to your liking ? Add comments if necessary.	Rating	1 - Website	4 - unknown	New OR 2- drink	Very Satisfied / Satisfied / No Opinion / Not Satisfied/ Very Unsatisfied	
Do you have any further comments or suggestions ?	Comment	1 - Website	4 - unknown			Free text - Very enjoyable experience.

File:https://secure.powertext.co.uk/SOAP/V2/Feedback.wsdl **Namespace:**FeedbackSOAP

10.2 Method: createFeedback

This function is used to send a contacts feedback data to the database.

10.2.1 Example: createFeedback

Function: `createFeedback(username As String, password As String, contactID As Integer, unitID As Integer, feedback_source_id As Integer, feedback_type_id As Integer, feedback_ratings As Object, feedback_comments As Object) As Integer`

.NET

```
        Dim myService As New FeedbackSOAP.Feedback
        Dim feedback_contact_id As String = "2099954"
        Dim feedback_unit_id As String = "67"
        Dim feedback_source_id As String = "4"
        Dim feedback_type_id As String = "1"

        Dim feedback_ratings(1) As FeedbackSOAP.feedbackRating

        Dim feedback_rating0 As New FeedbackSOAP.feedbackRating

        feedback_rating0.feedback_rating_category_idSpecified = True
        feedback_rating0.feedback_rating_category_id = 1
        feedback_rating0.feedback_rating_text = "Very Satisfied"
        feedback_rating0.feedback_note_text = "Nice curries"
        feedback_ratings(0) = feedback_rating0

        Dim feedback_rating1 As New FeedbackSOAP.feedbackRating

        feedback_rating1.feedback_rating_category_idSpecified = True
        feedback_rating1.feedback_rating_category_id = 3
        feedback_rating1.feedback_rating_text = "Very Satisfied"
        feedback_rating1.feedback_note_text = ""
        feedback_ratings(1) = feedback_rating1

        Dim feedback_comments(0) As FeedbackSOAP.feedbackComment

        Dim feedback_comment0 As New FeedbackSOAP.feedbackComment
        feedback_comment0.feedback_note_type = "contact"
        feedback_comment0.feedback_note_text = "The outside is a bit dated"
        feedback_comments(0) = feedback_comment0

        Dim feedback_id As Integer

        feedback_id = myService.createFeedback(username, password,
feedback_contact_id, feedback_unit_id, feedback_source_id, feedback_type_id,
feedback_ratings, feedback_comments)
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Feedback.wsdl";
    $client = new SoapClient($wsdl);
    $feedback_ratings = array(
        array('feedback_rating_category_id'=>'3',
            'feedback_rating_text'=>'Very Satisfied',
            'feedback_note_text'=>'Hot and Spicy just how I like my curries'),
        array('feedback_rating_category_id'=>'2',
            'feedback_rating_text'=>'Very Satisfied'
        )
    );

    $feedback_comments = array(
        array('feedback_note_type'=>'contact',
              'feedback_note_text'=>'The outside is a bit dated and will put
                  people off')
    );

    $feedback_contactID = '2099954';
    $feedback_unit_id='67';
    $feedback_source_id='4';
    $feedback_type_id='1';

    $myFeedbackID = $client->createFeedback("username", "password",
$feedback_contactID, $feedback_unit_id, $feedback_source_id, $feedback_type_id,
$feedback_ratings, $feedback_comments);

    if($myFeedbackID instanceof SoapFault) {
    echo "SOAP Fault: ".$myFeedbackID->faultstring;
    } elseif($myFeedbackID >= '1') {
    echo "Success, new feedback_id: ".$myFeedbackID;
    } else {
        echo "Failed to create feedback";
    }
```

RETURNS feedbackID As Integer   (feedbackID >= 1 for success)

10.3 Method: createFeedbackComment

Can be used to add client comments to a feedback instance.

10.3.1 Example: createFeedbackComment

Function: `createFeedbackComment(Username As String, Password As String, feedback_id As Integer, feedback_comments As Object) As Integer`

.NET

```
        Dim myService As New FeedbackSOAP.Feedback
        Dim feedback_comment_feedback_id As String = "40"
        Dim feedback_id As Integer

        Dim feedback_comments(0) As FeedbackSOAP.feedbackComment

        Dim feedback_comment0 As New FeedbackSOAP.feedbackComment

        feedback_comment0.feedback_note_type = "client"
        feedback_comment0.feedback_note_text = "We are having the facade of the
building upgraded in April. We will lose 2 parking spaces for 3 weeks. We hope
this does not cause too much inconvenience"
        feedback_comments(0) = feedback_comment0

        feedback_id = myService.createFeedbackComment("username", "password",
feedback_comment_feedback_id, feedback_comments)
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Feedback.wsdl";
    $client = new SoapClient($wsdl);
    $feedback_comments = array(
        array('feedback_note_type'=>'client',
              'feedback_note_text'=>'We consider these to be fair comments and
will act upon them as soon as possible')
    );

    $feedback_id = '40';

    $myFeedbackID = $client->createFeedbackComment("username", "password",
$feedback_id, $feedback_comments);

    if($myFeedbackID instanceof SoapFault) {
    echo "SOAP Fault: ".$myFeedbackID->faultstring;
    } elseif($myFeedbackID == $feedback_id) {
    echo  "Success, feedback_id: ".$myFeedbackID." updated";
    } else {
        echo "Failed to create feedback";
    }
```

RETURNS `feedbackID As Integer   (feedbackID the same as that passed for success)`

10.4 Method: getFeedback

Retrieves all ratings and comments for a given feedbackID.

10.4.1 Example: getFeedback

Function: `getFeedback(Username As String, Password As String, feedbackID As Integer) As FeedbackSOAP.Feedback`

.NET

```
    Try
        Dim myService As New FeedbackSOAP.Feedback
        Dim myFeedback As New FeedbackSOAP.feedbackStruct

        myFeedback = myService.getFeedback("username", "password", 40)

    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Feedback.wsdl";
    $client = new SoapClient($wsdl);
    $myFeedbackStruct = $client->getFeedback("username ","password",40);

    if($myFeedbackStruct instanceof SoapFault) {
    echo "SOAP Fault: ".$myFeedbackStruct->faultstring;
    } elseif(isset($myFeedbackStruct->feedback->feedback_contact_id)) {
        echo "Successfully received feedback data";
    } else {
        echo "Failed to get feedback";
    }
```

RETURNS `FeedbackSOAP.Feedback Object`

10.5 Method: getFeedbackContact

Retrieves all ratings and comments for a given feedback Contact. with output sort and rows options

10.5.1 Example: getFeedback

Function: `getFeedbackcontact (Username As String, Password As String, contactID As Integer, myResultOptions As Object) As FeedbackSOAP.Feedback()`

.NET

```
    Try
        Dim myService As New FeedbackSOAP.Feedback
        Dim myFeedback() As FeedbackSOAP.feedbackContactStruct
        Dim myResultOptions As New FeedbackSOAP.resultOptions

        Dim offset As Integer = 0
        Dim limit As Integer = 10

        myResultOptions.result_offset = "0"
        myResultOptions.result_limit = "10"
        myResultOptions.sort_by_element_name = "feedback_created_datetime"
        myResultOptions.sort_direction = "asc"

        myFeedback = myService.getFeedbackContact("username", "password",
2099954, myResultOptions)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Feedback.wsdl";
    $client = new SoapClient($wsdl);
    $myResultOptions['sort_by_element_name'] = 'feedback_created_datetime';
    $myResultOptions['sort_direction'] = 'asc';
    $myResultOptions['result_limit'] = '10';
    $myResultOptions['result_offset'] = '0';

    // load ConatctID 2099954 data
    $myFeedbackStruct = $client-
>getFeedbackContact("username","password",2099954,$myResultOptions);

    if($myFeedbackStruct instanceof SoapFault) {
        echo "SOAP Fault: ".$myFeedbackStruct->faultstring;
    } elseif(isset($myFeedbackStruct[0]->feedback_contact->feedback_id)) {
        echo "Successfully received feedback data";
    } else {
        echo "Failed to get feedback";
    }
```

RETURNS `FeedbackSOAP.Feedback()`

10.6 Method: getFeedbackCategories

Retrieves all categories used in the Feedback system.

10.6.1 Example: getFeedbackCategories

Function: `getFeedbackCategories(Username As String, Password As String) As FeedbackSOAP.FeedbackCategory()`

.NET

```
    Try
        Dim myService As New FeedbackSOAP.Feedback
        Dim myCategories() As FeedbackSOAP.feedbackCategory

        myCategories = myService.getFeedbackCategories("username", "password")
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Feedback.wsdl";
    $client = new SoapClient($wsdl);
    $myCategories = $client->getFeedbackCategories("username","password");

    if($myCategories instanceof SoapFault) {
        echo "SOAP Fault: ".$ myCategories ->faultstring;
    } elseif(count($myCategories) >= '1') {
        echo "Successfully received feedback categories data";
    } else {
        echo "Failed to get feedback categories";
    }
```

RETURNS `FeedbackSOAP.FeedbackCategory()`

10.7 Method: getFeedbackSources

Retrieves all sources used in the Feedback system.

10.7.1 Example: getFeedbackSources

Function: `getFeedbackSources (Username As String, Password As String) As FeedbackSOAP.FeedbackSources()`

.NET

```
    Try
        Dim myService As New FeedbackSOAP.Feedback
        Dim mySources() As FeedbackSOAP.feedbackSource

        mySources = myService.getFeedbackSources("username", "password")
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/Feedback.wsdl";
    $client = new SoapClient($wsdl);
    $myFeedbackSources = $client->getFeedbackSources("username ","password");

    if($myFeedbackSources instanceof SoapFault) {
        echo "SOAP Fault: ".$myFeedbackSources->faultstring;
    } elseif(isset($myFeedbackSources[0]->feedback_source_id)) {
        echo "Successfully received feedback source data";
    } else {
        echo "Failed to get feedback categories";
    }
```

RETURNS `FeedbackSOAP.FeedbackSource()`

10.8 Method: searchFeedback

This function can be used to fetch multiple feedback records in their basic form. Results may be sorted by a chosen field and offset limit values provide for paging. The maximum limit value will be restricted to 50 records per query.

10.8.1 Example: searchFeedback

Function: `searchFeedback(username As String, password As String,`

```
feedback_equal_arguments As FeedbackSOAP.feedbackObject,
feedback_greater_equal_arguments As FeedbackSOAP.feedbackObject,
feedback_less_equal_arguments As FeedbackSOAP.feedbackObject,
feedback_in_sql_arguments As FeedbackSOAP.feedbackObjectInSQL,
result_options As FeedbackSOAP.resultOptions) As FeedbackSOAP.feedbackStruct()
```

.NET

```
Try

    Dim myService As New FeedbackSOAP.Feedback
    Dim myFeeback() As FeedbackSOAP.feedbackStruct
    Dim myFeedbackEqualsArgs As New FeedbackSOAP.feedbackObject
    Dim myFeedbackGreaterThanArgs As New FeedbackSOAP.feedbackObject
    Dim myFeedbackLessThanArgs As New FeedbackSOAP.feedbackObject
    Dim myFeedbackInSQLArgs As New FeedbackSOAP.feedbackObjectInSQL
    Dim myResultOptions As New FeedbackSOAP.resultOptions

    ' restrict to unitID 67
    myFeedbackEqualsArgs.feedback_unit_id = 67
    myFeedbackEqualsArgs.feedback_unit_idSpecified = True

    ' set filter to be enquiries in  month of "March 2013"
    myFeedbackGreaterThanArgs.feedback_enquiry_datetime = "2013-03-01 00:00:00"
    myFeedbackLessThanArgs.feedback_enquiry_datetime = "2013-03-31 23:59:59"

    myFeedback = myService.searchFeedback("username","password",
myFeedbackEqualsArgs,
myFeedbackGreaterThanArgs,
 myFeedbackLessThanArgs,
myFeedbackInSQLArgs, myResultOptions)

  If myFeedback.Length < 1 Then
        MsgBox("Failed to get any feedback results.")
    Else
        MsgBox("Success, found "&myFeedback.Length &" records, "&
"First result feedback_id: " &
myFeedback(0).feedback.feedback_id)
    End If

Catch ex As Exception
    MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$client = new SoapClient("../Feedback.wsdl",array("trace"=>1,"exceptions"=>0));

$myFeedbackEqualsArgs = array(); // exact match args

// restrict to unitIDs 67,68 and 69
$myFeedbackInSQLArgs['feedback_unit_id'] = '67,68,69';

// set filter to be enquiries in  month of "March 2013"
$myFeedbackGreaterThanArgs['feedback_enquiry_datetime'] = "2013-03-01 00:00:00";
$myFeedbackLessThanArgs['feedback_enquiry_datetime'] = "2013-03-31 23:59:59";

// add some extra result option arguments
$myResultOptions['sort_by_element_name'] = 'feedback_enquiry_datetime';
$myResultOptions['sort_direction'] = 'desc';
$myResultOptions['result_limit'] = '10';
$myResultOptions['result_offset'] = '0';

// run search
$myFeedback = $client->searchFeedback("username","password",
$myFeedbackEqualsArgs,
$myFeedbackGreaterThanArgs,
$myFeedbackLessThanArgs,
$myFeedbackInSQLArgs,
$myResultOptions);

if($myFeedback instanceof SoapFault) {
    echo "SOAP Fault: ".$myFeedback->faultstring;
} elseif(count($myFeedback) >= '1') {
    echo "Success, found: ".count($myFeedback);
} else {
    echo "Failed to find any feedback for given criteria";
}
```

RETURNS

```
Object As FeedbackSOAP.feedbackStruct()
```

10.9 Method: searchFeedbackCount

This function can be used to return a total count of feedback records available via a searchFeedback() call.

10.9.1 Example: searchFeedbackCount

Function: `searchFeedbackCount(username As String, password As String, feedback_equal_arguments As FeedbackSOAP.feedbackObject, feedback_greater_equal_arguments As FeedbackSOAP.feedbackObject, feedback_less_equal_arguments As FeedbackSOAP.feedbackObject,feedback_in_sql_arguments As FeedbackSOAP.feedbackObjectInSQL, result_options As FeedbackSOAP.resultOptions) count As Integer`

.NET

```
Try

    Dim myService As New FeedbackSOAP.Feedback
    Dim myCount As Integer
    Dim myFeedbackEqualsArgs As New FeedbackSOAP.feedbackObject
    Dim myFeedbackGreaterThanArgs As New FeedbackSOAP.feedbackObject
    Dim myFeedbackLessThanArgs As New FeedbackSOAP.feedbackObject
    Dim myFeedbackInSQLArgs As New FeedbackSOAP.feedbackObjectInSQL
    Dim myResultOptions As New FeedbackSOAP.resultOptions

    ' restrict to unitID 67
    myFeedbackEqualsArgs.feedback_unit_id = 67
    myFeedbackEqualsArgs.feedback_unit_idSpecified = True

    ' set filter to be enquiries in  month of "March 2013"
    myFeedbackGreaterThanArgs.feedback_enquiry_datetime = "2013-03-01 00:00:00"
    myFeedbackLessThanArgs.feedback_enquiry_datetime = "2013-03-31 23:59:59"

    myCount = myService.searchFeedbackCount("username","password",
myFeedbackEqualsArgs,
myFeedbackGreaterThanArgs,
 myFeedbackLessThanArgs,
myFeedbackInSQLArgs, myResultOptions)

  If myCount >= 1 Then
        MsgBox("Found " & myCount & " feedbacks.")
    Else
        MsgBox("No feedbacks found.")
    End If

Catch ex As Exception
    MsgBox("Exception: " + ex.Message)
End Try
```

.PHP

```
$client = new SoapClient("../Feedback.wsdl",array("trace"=>1,"exceptions"=>0));

$myFeedbackEqualsArgs = array(); // exact match args

// restrict to unitIDs 67,68 and 69
$myFeedbackInSQLArgs['feedback_unit_id'] = '67,68,69';

// set filter to be enquiries in  month of "March 2013"
$myFeedbackGreaterThanArgs['feedback_enquiry_datetime'] = "2013-03-01 00:00:00";
$myFeedbackLessThanArgs['feedback_enquiry_datetime'] = "2013-03-31 23:59:59";

// add some extra result option arguments
$myResultOptions['sort_by_element_name'] = 'feedback_enquiry_datetime';
$myResultOptions['sort_direction'] = 'desc';
$myResultOptions['result_limit'] = '10';
$myResultOptions['result_offset'] = '0';

// run search
$myCount = $client->searchFeedbackCount("username","password",
$myFeedbackEqualsArgs,
$myFeedbackGreaterThanArgs,
$myFeedbackLessThanArgs, $myResultOptions);

if($myCount instanceof SoapFault) {
    echo "SOAP Fault: ".$myCount->faultstring;
} elseif($myCount >= '1') {
    echo "Success, found: ". $myCount;
} else {
    echo "Failed to find any feedback records for given criteria";
}
```

RETURNS

```
Count As Integer
```

11. Wifi Interaction API

The Wifi Interaction API provides methods for saving Wifi Interactions.

File:https://secure.powertext.co.uk/SOAP/V2/WifiInteraction.wsdl **Namespace:**WifiInteractionSOAP

11.1 Method: createWifiInteraction

Can be used to save Wifi interaction data against a contact record.

11.1.1 Example: createWifiInteraction

Function: createWifiInteraction(Username As String, Password As String, hotspot_name As String, mac_address As String, device_mime_type As String, interaction_type As String, contactID As Integer) As Integer

.NET

```
    Try
        Dim myService As New WifiInterationSOAP.WifiInteraction
        Dim mywifiinteraction_id As Integer

       'pass 1 as contact_id if not known
        mywifiinteraction_id = myService.createWifiInteraction("username",
"password", "Revolution top floor", "00:A0:C9:14:C8:29", "powertextapp",
"detected", 1)

        mywifiinteraction_id = myService.createWifiInteraction("username ",
"password", "Revolution top floor", "00:A0:C9:14:C8:29", "powertextapp",
"connected", 1)

        mywifiinteraction_id = myService.createWifiInteraction("username ",
"password", "Revolution top floor", "00:A0:C9:14:C8:29", "powertextapp",
"authenticated", 2099954)

   End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/WifiInteraction.wsdl";
    $client = new SoapClient($wsdl);
    $wifiinteraction_hotspot_name = 'Revolution top floor';
    $wifiinteraction_mac_name='00:A0:C9:14:C8:29';
    $wifiinteraction_device_mime_type='powertextapp';
    $wifiinteraction_interaction_type='detected';
    $wifiinteraction_contact_id=1;

    $myWifiInteractionID = $client-
>createWifiInteraction("username","password",$wifiinteraction_hotspot_name,$wifi
interaction_mac_name,$wifiinteraction_device_mime_type,$wifiinteraction_interact
ion_type,$wifiinteraction_contact_id);


    if($myWifiInteractionID instanceof SoapFault) {
    echo "SOAP Fault: ".$myWifiInteractionID->faultstring;
    } elseif($myWifiInteractionID >= '1') {
    echo "Success, new wifiinteraction_id: ".$myWifiInteractionID;
    } else {
        echo "Failed to create wifiinteraction";
    }
```

RETURNS mywifiinteractionID As Integer   (mywifiinteractionID >= 1 for success)

12. EPOS API

The EPOS API provides methods for saving point of sale transactions
File:https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl **Namespace:**EPOSSOAP

12.1 Method: createPurchaseHistory

Can be used to save purchase information for a contact.

12.1.1 Example: createPurchaseHistory

Function:
```
createPurchaseHistory(Username As String, Password As String,
transaction_reference As String,  contactID As Integer, unitID As Integer,
transaction_timestamp As Integer, transaction_lines As Object)
purchase_history_id As Integer
```

.NET

```
    Try
      Dim myService As New EPOSSOAP.EPOS
      Dim purchase_history_transaction_reference As String = "The Cricket Inn"
      Dim purchase_history_contact_id As String = "2099954"
      Dim purchase_history_unit_id As String = "67"
      Dim purchase_history_transaction_timestamp As Long =
CLng(DateTime.UtcNow.Subtract(New DateTime(1970, 1, 1)).TotalSeconds)
      Dim purchase_history_transaction_lines(1) As
EPOSSOAP.purchaseHistoryTransactionLines
      Dim purchase_history_transaction_line0 As New
EPOSSOAP.purchaseHistoryTransactionLines
      purchase_history_transaction_line0.purchase_history_transaction_line =
"White Swan"
      purchase_history_transaction_line0.purchase_history_transaction_line_sku =
"123456789012345678901234567890123456"
      purchase_history_transaction_line0.purchase_history_transaction_line_quant
ity = "2"

      purchase_history_transaction_line0.purchase_history_transaction_line_value
= "5.6"
      purchase_history_transaction_lines(0) = purchase_history_transaction_line0
      Dim purchase_history_transaction_line1 As New
EPOSSOAP.purchaseHistoryTransactionLines
      purchase_history_transaction_line1.purchase_history_transaction_line =
"Kipling"
      purchase_history_transaction_line_sku=>'12345678901234567890123456789012345
56',
      purchase_history_transaction_line0.purchase_history_transaction_line_sku =
"123456789012345678901234567890123455"

      purchase_history_transaction_line1.purchase_history_transaction_line_quant
ity = "2"

      purchase_history_transaction_line1.purchase_history_transaction_line_value
= "6.4"
      purchase_history_transaction_lines(1) = purchase_history_transaction_line1

      Dim purchase_history_id As Integer
      purchase_history_id = myService.createPurchaseHistory("username",
"password", purchase_history_transaction_reference, purchase_history_contact_id,
purchase_history_unit_id, purchase_history_transaction_timestamp,
purchase_history_transaction_lines)

End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl";
    $client = new SoapClient($wsdl);
    $purchase_history_transaction_lines = array(
        array('purchase_history_transaction_line'=>'White Swan',
      'purchase_history_transaction_line_sku'=>'12345678901234567890123456789012
3456',
        'purchase_history_transaction_line_quantity'=>1,
        'purchase_history_transaction_line_value'=>2.80),
        array('purchase_history_transaction_line'=>'Kipling',
      'purchase_history_transaction_line_sku'=>'12345678901234567890123456789012
3455',
        'purchase_history_transaction_line_quantity'=>1,
        'purchase_history_transaction_line_value'=>3.20)
        );


    $purchase_history_transaction_reference = 'Hallamshire';
    $purchase_history_contact_id='2099954';
    $purchase_history_unit_id='67';
    $purchase_history_transaction_timestamp=`date +%s`;
    $myPurchaseHistoryID = $client-
>createPurchaseHistory("username","password",$purchase_history_transaction_refer
ence,$purchase_history_contact_id,$purchase_history_unit_id,$purchase_history_tr
ansaction_timestamp,$purchase_history_transaction_lines);
```

RETURNS `purchase_history_id As Integer`

12.2 Method: updatePurchaseHistory

Can be used to update existing purchase information for a single purchase by a contact. All transaction lines for the purchase will be deleted and re-inserted so transaction lines must be a complete record.

12.2.1 Example: updatePurchaseHistory

Function: `updatePurchaseHistory(Username As String, Password As String, purchase_history_id As Integer, transaction_reference As String, contactID As Integer, unitID As Integer, transaction_timestamp As Integer, transaction_lines As Object) purchase_history_id As Integer`

.NET

```
    Try
      Dim myService As New EPOSSOAP.EPOS
        Dim purchase_history_id As String = "20"
        Dim purchase_history_transaction_reference As String = "The Cricket Inn"
        Dim purchase_history_contact_id As String = "2099954"
        Dim purchase_history_unit_id As String = "67"
        Dim purchase_history_transaction_timestamp As Long =
CLng(DateTime.UtcNow.Subtract(New DateTime(1970, 1, 1)).TotalSeconds)
```

```
        Dim purchase_history_transaction_lines(1) As
EPOSSOAP.purchaseHistoryTransactionLines
        Dim purchase_history_transaction_line0 As New
EPOSSOAP.purchaseHistoryTransactionLines
        purchase_history_transaction_line0.purchase_history_transaction_line =
"White Swan"
        purchase_history_transaction_line0.purchase_history_transaction_line_sku
= "12345678901234567890123456789012 3456"

purchase_history_transaction_line0.purchase_history_transaction_line_quantity =
"1"

purchase_history_transaction_line0.purchase_history_transaction_line_value =
"2.8"
        purchase_history_transaction_lines(0) =
purchase_history_transaction_line0

        Dim purchase_history_transaction_line1 As New
EPOSSOAP.purchaseHistoryTransactionLines
        purchase_history_transaction_line1.purchase_history_transaction_line =
"Kipling"
        purchase_history_transaction_line1.purchase_history_transaction_line_sku
= "12345678901234567890123456789012 3455"

purchase_history_transaction_line1.purchase_history_transaction_line_quantity =
"2"

purchase_history_transaction_line1.purchase_history_transaction_line_value =
"6.4"
        purchase_history_transaction_lines(1) =
purchase_history_transaction_line1

        purchase_history_id = myService.updatePurchaseHistory("username",
"password", purchase_history_id, purchase_history_transaction_reference,
purchase_history_contact_id, purchase_history_unit_id,
purchase_history_transaction_timestamp, purchase_history_transaction_lines)
```

End Try

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl";
    $client = new SoapClient($wsdl);
    $purchase_history_transaction_lines = array(
        array('purchase_history_transaction_line'=>'White Swan Ale',
      'purchase_history_transaction_line_sku'=>'12345678901234567890123456789012
3456',
        'purchase_history_transaction_line_quantity'=>2,
        'purchase_history_transaction_line_value'=>5.60),
        array('purchase_history_transaction_line'=>'Kipling Ale',
      'purchase_history_transaction_line_sku'=>'12345678901234567890123456789012
3456',
        'purchase_history_transaction_line_quantity'=>2,
        'purchase_history_transaction_line_value'=>6.40)
        );

    $purchase_history_id='19';
    $purchase_history_transaction_reference = 'Hallamshire Pub';
    $purchase_history_contact_id='2099954';
    $purchase_history_unit_id='67';
    $purchase_history_transaction_timestamp=`date +%s`;
    $myPurchaseHistoryID = $client-
>updatePurchaseHistory("username","password",$purchase_history_id,$purchase_hist
ory_transaction_reference,$purchase_history_contact_id,$purchase_history_unit_id
,$purchase_history_transaction_timestamp,$purchase_history_transaction_lines);
```

RETURNS number of rows updated. Please note that 1 master row will be updated. All transaction lines will be deleted and then inserted. So if you delete a purchase with 3 transaction lines and update with 4 then the result will be 1 + 3 + 4 = 8.

12.3 Method: deletePurchaseHistory

Can be used to delete existing purchase information.

12.3.1 Example: deletePurchaseHistory

Function: `deletePurchaseHistory(Username As String, Password As String, purchase_history_id As Integer)`

.NET

```
    Try
      Dim myService As New EPOSSOAP.EPOS
      Dim purchase_history_id As String = "11"
      Dim rowsdeleted As Integer
      rowsdeleted = myService.deletePurchaseHistory("username", "password",
purchase_history_id)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl";
    $client = new SoapClient($wsdl);
    $purchase_history_id='10';
    $mydeleterows = $client-
>deletePurchaseHistory("username","password",$purchase_history_id);
```

RETURNS Number of rows deleted. They should be 1 row for the master tables and 1 row for each transaction line.

12.4 Method: getPurchaseHistoryPurchase

Can be used to return all data for a single purchase.

12.4.1 Example: getPurchaseHistoryPurchase

Function: `getPurchaseHistoryPurchase(Username As String, Password As String, purchase_history_id As Integer) As Object`

.NET

```
    Try
      Dim myService As New EPOSSOAP.EPOS
      Dim purchase_history_id As String = "21"
      Dim myPurchaseHistory As EPOSSOAP.purchaseStruct
       myPurchaseHistory = myService.getPurchaseHistoryPurchase("username",
"password", purchase_history_id)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl";
    $client = new SoapClient($wsdl);
    $purchase_history_id='19';
    $myPurchaseHistoryObj = $client-
>getPurchaseHistoryPurchase("username","password",$purchase_history_id);
```

RETURNS `PurchaseHistoryObject`

12.5 Method: getPurchaseHistoryContact

Can be used to return all purchase data for a contact.

12.5.1 Example: getPurchaseHistoryContact

Function: `getPurchaseHistoryPurchase(Username As String, Password As String, contactID As Integer) As Object`

.NET

```
    Try
      Dim myService As New EPOSSOAP.EPOS
      Dim purchase_contact_id As String = "21"
      Dim myPurchaseHistory() As EPOSSOAP.purchaseStruct
      myPurchaseHistory = myService.getPurchaseHistoryContact("username",
"password", purchase_contact_id)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl";
    $client = new SoapClient($wsdl);
    $purchase_contact_id='2099954';
    $myPurchaseHistoryObj = $client-
>getPurchaseHistoryContact("username","password",$purchase_contact_id);
```

RETURNS `PurchaseHistoryObject()`

12.6 Method: getPurchaseHistoryUnit

Can be used to return all purchase data for a unit.

12.6.1 Example: getPurchaseHistoryUnit

Function: `getPurchaseHistoryUnit(Username As String, Password As String, unitID As Integer) As Object`

.NET

```
    Try
      Dim myService As New EPOSSOAP.EPOS
      Dim purchase_unit_id As String = "21"
      Dim myPurchaseHistory() As EPOSSOAP.purchaseStruct
      myPurchaseHistory = myService.getPurchaseHistoryUnit("username",
"password", purchase_unit_id)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl";
    $client = new SoapClient($wsdl);
    $purchase_unit_id='67';
    $myPurchaseHistoryObj = $client-
>getPurchaseHistoryUnit("username","password",$purchase_unit_id);
```

RETURNS `PurchaseHistoryObject()`

12.7 Method: getPurchaseHistoryReference

Can be used to return all data for a transaction reference.

12.7.1 Example: getPurchaseHistoryReference

Function: `getPurchaseHistoryUnit(Username As String, Password As String, transaction_reference As String) As Object`

.NET

```
    Try
      Dim myService As New EPOSSOAP.EPOS
      Dim purchase_transaction_reference As String = "Greystoke"
      Dim myPurchaseHistory() As EPOSSOAP.purchaseStruct
      myPurchaseHistory = myService.getPurchaseHistoryReference("username",
"password", purchase_transaction_reference)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/EPOS.wsdl";
    $client = new SoapClient($wsdl);
    $purchase_transaction_reference ='Greystoke';
    $myPurchaseHistoryObj = $client-
>getPurchaseHistoryReference("username","password",$purchase_transaction_referen
ce);
```

RETURNS `PurchaseHistoryObject()`

13. UnitUnsubscribe API

The Unit Unsubscribe API provides methods for unsubscribing from individual units. Head Office will not be included in any data returned. This API will not allow unsubscription from Head Office. However if all other units have been unsubscribed then the Email Allow flag is set to No for the contact and so all Eflyers are disabled.

File:https://secure.powertext.co.uk/SOAP/V2/UnitUnsubscribe.wsdl **Namespace:**UnitUnsubscribe

13.1 Method: getGroupUnits

Returns an array of units that the contact has a group membership in.

13.1.1 Example: getGroupUnits

Function: `getGroupUnits(Username As String, Password As String, contactID As Integer) UnitData As Array`

.NET

```
    Try
      Dim contact_id As Integer = 47069
      Dim myUnitData() As UnitUnsubscribe.UnitData
      myUnitData = myService.getGroupUnits("username", "password", contact_id)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/UnitUnsubscribe.wsdl";
    $client = new SoapClient($wsdl);
    $contact_id='47069';
    $myUnitData = $client->getGroupUnits("username","password",$contact_id);
```

RETURNS `myUnitData() array of unitid, unitname, subscribed`

13.2 Method: getUnitSubscriptions

Returns an array of units that the contact is subscribed to.

13.2.1 Example: getUnitSubscriptions

Function: `getUnitSubscriptions(Username As String, Password As String, contactID As Integer) UnitData As Array`

.NET

```
    Try
      Dim contact_id As Integer = 47069
      Dim myUnitData () As UnitUnsubscribe.UnitData
      myUnitData = myService.getUnitSubscriptions ("username", "password",
contact_id)
```

```
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/UnitUnsubscribe.wsdl";
    $client = new SoapClient($wsdl);
    $contact_id='47069';
    $myUnitData = $client->
getUnitSubscriptions("username","password",$contact_id);
```

RETURNS `myUnitData() array of unitid, unitname, subscribed`

13.3 Method: getUnitLastInteraction

Returns unit that the contact last interacted with. The API returns the last non Head Office unit interacted with. If no units are returned then the API checks the Head Office interactions and will return that unit if interactions exist.

13.3.1 Example: getUnitLastInteraction

Function: `getUnitLastInteraction (Username As String, Password As String, contactID As Integer) UnitData As Array`

.NET

```
    Try
      Dim contact_id As Integer = 47069
      Dim myUnitData () As UnitUnsubscribe.UnitData
      myUnitData = myService. getUnitLastInteraction("username", "password",
contact_id)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/UnitUnsubscribe.wsdl";
    $client = new SoapClient($wsdl);
    $contact_id='47069';
    $myUnitData = $client->
getUnitLastInteraction("username","password",$contact_id);
```

RETURNS `myUnitData() array of unitid, unitname, subscribed`

13.4 Method: unsubscribeContact

Unsubscribe a contact from a unit. The API will remove the contact from all groups in that Unit and then stop broadcasts from that unit to the contact. If all units have been unsubscribed from then all Eflyers are stopped.

13.4.1 Example: unsubscribeContact

Function: `unsubscribeContact (Username As String, Password As String, unitID as Integer, contactID As Integer, reason as String) UnitData As Array`

.NET

```
    Try
      Dim contact_id As Integer = 47069
      Dim unit_id As Integer = 999
      Dim reason As String = 'Emigrating'
      Dim myUnitData () As UnitUnsubscribe.UnitData
      myUnitData = myService. unsubscribeContact ("username", "password",
unit_id, contact_id,reason)
    End Try
```

.PHP

```
    $wsdl = "https://secure.powertext.co.uk/SOAP/V2/UnitUnsubscribe.wsdl";
    $client = new SoapClient($wsdl);
    $contact_id='47069';
    $unit_id='999';
    $reason='Emigrating';
    $ myUnitData = $client->
unsubscribeContact("username","password",$unit_id,$contact_id,$reason);
```

RETURNS `myUnitData() array of unitid, unitname, subscribed`

14. RAW XML Request Examples

The following examples are raw XML requests, for those people who can't or do not want to use a pre-existing library or module. It is worth noting the date format required for fields of Date type. These must be passed through in YYYY-MM-DD format, this includes UDF fields of type Date.

14.1 unsubscribe Contact

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:ns1="https://secure.powertext.co.uk/SOAP"
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/"
                  SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
 <ns1:unsubscribeContact>
    <username xsi:type="xsd:string">yourusername</username>
    <password xsi:type="xsd:string">yourpassword</password>
    <unsubscribeContactRequest SOAP-ENC:arrayType="xsd:string[2]"
xsi:type="SOAP-ENC:Array">
     <item xsi:type="xsd:string">447911111111</item>
     <item xsi:type="xsd:string">me@example.com</item>
    </unsubscribeContactRequest>
 </ns1:unsubscribeContact>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```