



Pay24-7 Web Service Overview

Alexander Banthien
August 2005

Introduction and overview to the Pay24-7 web service.

Version: 0.4

Reviewers:

Table of Contents

1 Introduction.....	3
2 WSDL – Web Service Description	3
3 Payment Reporting By Identifier	3
3.1 Idea.....	3
3.2 Return Codes.....	4
3.3 Examples.....	4
4 Payment Reporting by Date.....	7
4.1 Idea.....	7
5 Balance Reporting.....	7
5.1 Idea.....	7
6 User Management.....	8
6.1 Idea.....	8
6.2 Return Codes.....	8
6.3 Pitfalls.....	9
6.4 Examples.....	11

1 Introduction

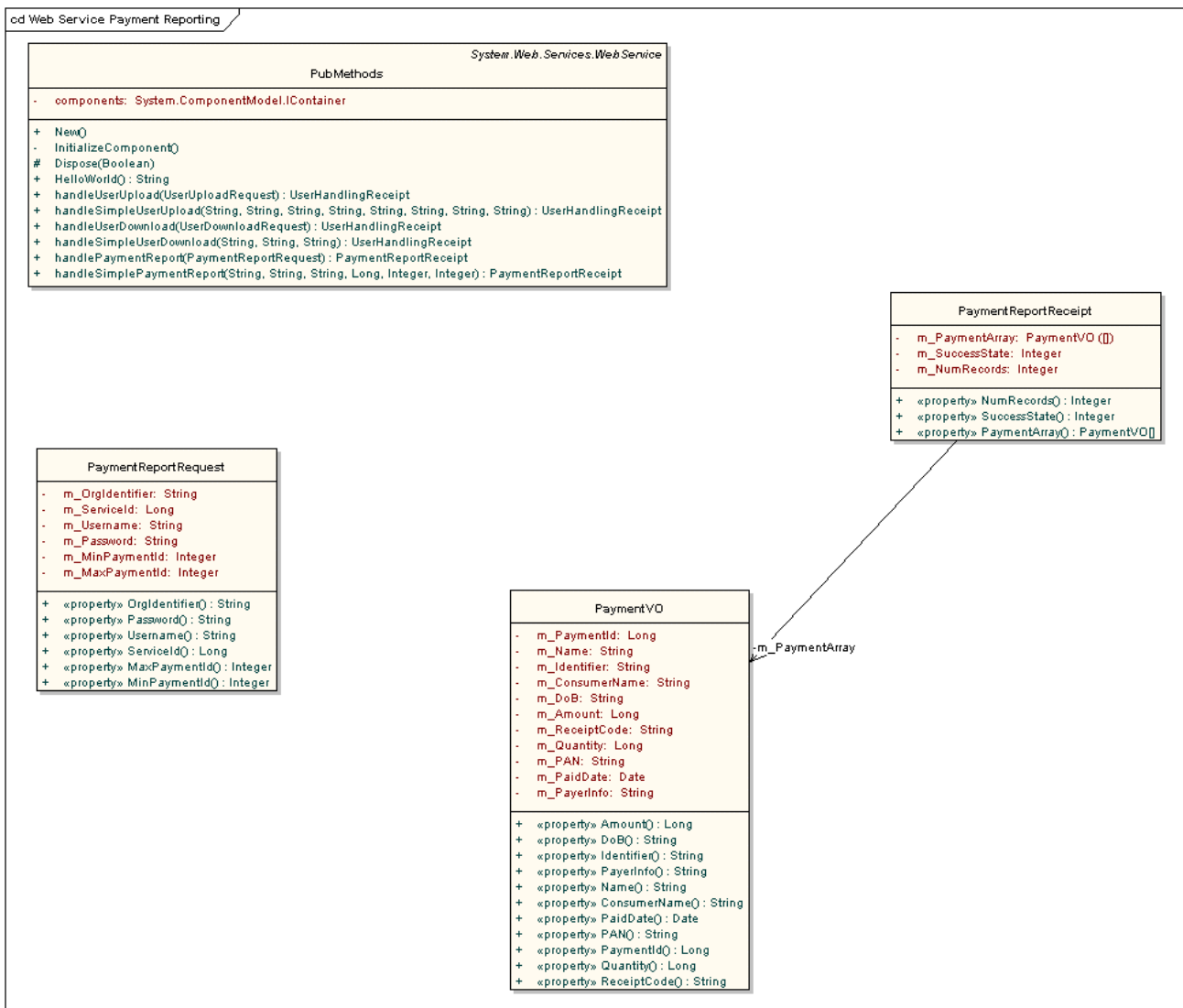
Pay24-7 publishes a web service to allow enabled client software to perform and automate several tasks.

At the time of this writing methods for the management of users and for the reporting of payments exist.

2 WSDL – Web Service Description

The URL for an up-to-date version of the WSDL file, please contact P247 staff at info@pay24-7.com.

3 Payment Reporting By Identifier



3.1 Idea

All access to the system happens through class PubMethods, which publishes several

methods.

The principle behind payment reporting is the following:

- client system and P247 share a common identifier for the users
- an identifier of a “service”, a receivable, is copied from P247 into the client
- a user account with sufficient privileges to download payment records has been set up

P247 allocates ascending integers (with gaps!) to the single transactions (i.e. To the single items within one aggregated payment transaction). These ids are reported by the web service as PaymentId in the PaymentVO class.

A client application typically stores these payment ids and sends the highest yet known paymentid along with the next request to retrieve only those records that are newer since the last download.

3.2 Return Codes

Any non 0 return code indicates a failure. Currently there are two possible codes:

Code	Meaning
1	Authentication failure: the username and password do not allow access to the requested resources. This may be wrong credentials or insufficient privileges.
2	Processing failure: this indicates an internal processing error, which we kindly ask you to report to us.

It should be noted here that handleSimplePaymentReport and handlePaymentReport are the same method, the prior simply calling the latter.

3.3 Examples

Accessing the API for payment reporting is usually quite simple. Example code that does it would look similar to the following (example given in VB.Net):

```
Dim prx As New P247WS.localhost.PubMethods
prx.Url = "https://www.pay24-7.com/P247WS/PubMethods.asmx"

Dim req As New P247WS.localhost.PaymentReportRequest
req.MaxPaymentId = -1 ' get 'em all
req.MinPaymentId = 0 ' get 'em all
req.OrgIdentifier = "myOrg"
req.Password = "verySecret"
req.ServiceId = srv1Id
req.Username = "myUserName"

Dim res As P247WS.localhost.PaymentReportReceipt
res = prx.handlePaymentReport(req)
```

this code creates the following XML request that gets sent across the internet:

3 Payment Reporting By Identifier

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handlePaymentReport xmlns="http://www.pay24-
7.com/P247WS/PubMethods">
      <_req>
        <OrgIdentifier>
          myOrg</OrgIdentifier>
        <Password>verySecret</Password>
        <Username>myUsername</Username>
        <ServiceId>5446</ServiceId>
        <MaxPaymentId>-1</MaxPaymentId>
        <MinPaymentId>0</MinPaymentId>
      </_req>
    </handlePaymentReport>
  </soap:Body>
</soap:Envelope>
```

The server sends back a response that – using a web service enabled programming language and development environment translates right back into client side object. The response XML looks like the following:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handlePaymentReportResponse
      xmlns="http://www.pay24-7.com/P247WS/PubMethods">
      <handlePaymentReportResult>
        <NumRecords>10</NumRecords>
        <SuccessState>0</SuccessState>
        <PaymentArray>
          <PaymentVO>
            <Amount>1000</Amount>
            <DoB>01/01/0001 00:00:00</DoB>
            <Identifier>1</Identifier>
            <ConsumerName> 1</ConsumerName>
            <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
            <PaymentId>70273</PaymentId>
            <Quantity>0</Quantity>
            <ReceiptCode>TEST REC CODE</ReceiptCode>
          </PaymentVO>
          <PaymentVO>
            <Amount>2000</Amount>
            <DoB>01/01/0001 00:00:00</DoB>
            <Identifier>2</Identifier>
            <ConsumerName> 2</ConsumerName>
            <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
            <PaymentId>70275</PaymentId>
            <Quantity>0</Quantity>
            <ReceiptCode>TEST REC CODE</ReceiptCode>
          </PaymentVO>
          <PaymentVO>
            <Amount>1000</Amount>
            <DoB>01/01/0001 00:00:00</DoB>
            <Identifier>3</Identifier>
            <ConsumerName> 3</ConsumerName>
```

3 Payment Reporting By Identifier

```

                                <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                <PaymentId>70277</PaymentId>
                                <Quantity>0</Quantity>
                                <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                <PaymentVO>
                                    <Amount>3000</Amount>
                                    <DoB>01/01/0001 00:00:00</DoB>
                                    <Identifier>4</Identifier>
                                    <ConsumerName> 4</ConsumerName>
                                    <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                <PaymentId>70279</PaymentId>
                                <Quantity>0</Quantity>
                                <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                <PaymentVO>
                                    <Amount>4000</Amount>
                                    <DoB>01/01/0001 00:00:00</DoB>
                                    <Identifier>5</Identifier>
                                    <ConsumerName> 5</ConsumerName>
                                    <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                <PaymentId>70281</PaymentId>
                                <Quantity>0</Quantity>
                                <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                <PaymentVO>
                                    <Amount>1100</Amount>
                                    <DoB>01/01/0001 00:00:00</DoB>
                                    <Identifier>6</Identifier>
                                    <ConsumerName> 6</ConsumerName>
                                    <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                <PaymentId>70283</PaymentId>
                                <Quantity>0</Quantity>
                                <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                <PaymentVO>
                                    <Amount>2300</Amount>
                                    <DoB>01/01/0001 00:00:00</DoB>
                                    <Identifier>7</Identifier>
                                    <ConsumerName> 7</ConsumerName>
                                    <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                <PaymentId>70285</PaymentId>
                                <Quantity>0</Quantity>
                                <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                <PaymentVO>
                                    <Amount>2000</Amount>
                                    <DoB>01/01/0001 00:00:00</DoB>
                                    <Identifier>8</Identifier>
                                    <ConsumerName> 8</ConsumerName>
                                    <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                <PaymentId>70287</PaymentId>
                                <Quantity>0</Quantity>
                                <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                <PaymentVO>
                                    <Amount>21000</Amount>

```

3 Payment Reporting By Identifier

```
                                <DoB>01/01/0001 00:00:00</DoB>
                                <Identifier>9</Identifier>
                                <ConsumerName> 9</ConsumerName>
                                <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                <PaymentId>70289</PaymentId>
                                <Quantity>0</Quantity>
                                <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                <PaymentVO>
                                    <Amount>300</Amount>
                                    <DoB>01/01/0001 00:00:00</DoB>
                                    <Identifier>10</Identifier>
                                    <ConsumerName> 10</ConsumerName>
                                    <PaidDate>2005-06-
06T15:59:00.0000000+01:00</PaidDate>
                                    <PaymentId>70291</PaymentId>
                                    <Quantity>0</Quantity>
                                    <ReceiptCode>TEST REC CODE</ReceiptCode>
                                </PaymentVO>
                                </PaymentArray>
                            </handlePaymentReportResult>
                        </handlePaymentReportResponse>
                    </soap:Body>
</soap:Envelope>
```

Here is should be observed that:

- The Amount fields are given in pence rather than pounds. A value of 1000 corresponds to 10 GBP not 1000.
- The Identifier is the personal identifier that was used when creating the user.
- Quantity is reported as 0 if the item is not quantifiable.

4 Payment Reporting by Date

4.1 Idea

Reporting by date via method `HandleXSettlementReport(..)` publishes more information on the individual payment records and reports on payments not only on one but across all services.

5 Balance Reporting

5.1 Idea

Pay24-7 internally maintains a balance over all payments that have been recorded on the system on any given service by all individual users.

It is possible to report on the balances by querying method `HandleXBalanceReport(..)`. It is mandatory to filter the query either by User identifier, or by Service Id or by both.

The `PaidTotal` field in the the entries of the `BalanceArray` member of the `BalanceReportReceipt` contains the balance.

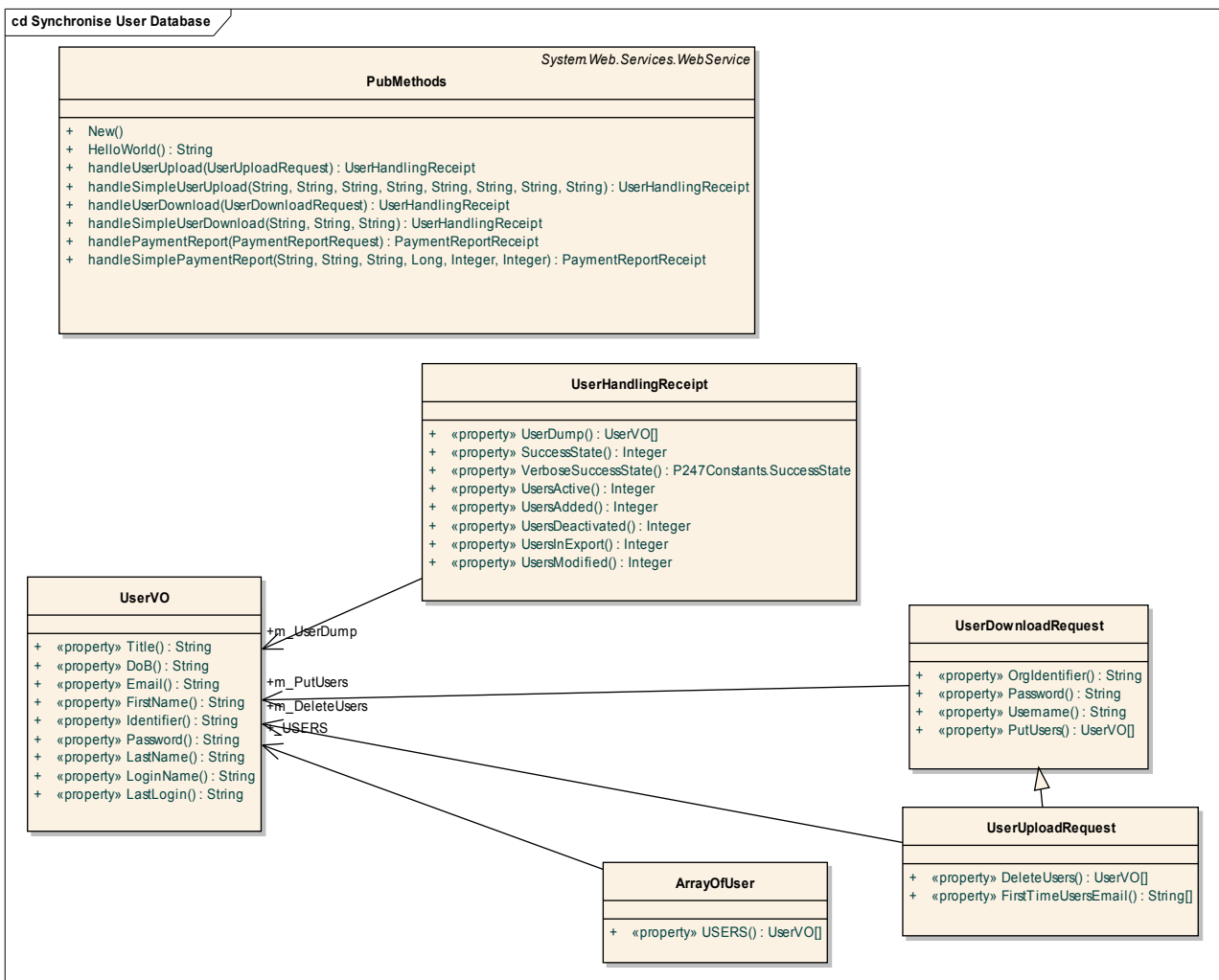
6 User Management

6.1 Idea

The user management methods allow creation, deletion (actually deactivation) and modification of user records.

All user records are identified by the Identifier field in the UserVO (value object, used for transfer across the net). Depending on whether a particular UserVO is in the array PutUser or DeleteUser in the UserDownloadRequest the get created/modified or deactivated in P247 respectively.

When requesting the download of just a particular set of users with their data, it is sufficient to populate the UserVO objects with the Identifier field when sending the UserDownloadRequest. Complete records will be sent back in the Receipt.



6.2 Return Codes

Any non 0 return code indicates a failure. Currently there are two possible codes:

Code	Meaning
1	Authentication failure: the username and password do not allow access to the requested resources. This may be wrong credentials or insufficient privileges.
2	Processing failure: this indicates an internal processing error, which we kindly as you to report to us.

It should be noted here that the “Simple” versions of the methods are generally not as rich as their full featured counterparts.

6.3 Pitfalls

A few remarks ought to be made regarding usage scenarios

Passwords

It is usually unwise to overwrite passwords as the users may have logged into P247 and changed their passwords.

When initially creating a user, it may be a good idea to set a – default – password and communicate that to the user.

When creating users the password may be left blank, in which case P247 creates passwords.

Once a password is set by either the user on the web or the web service, it is stored in hashed form in the database and cannot be retrieved. It can only be reset if it has been forgotten.

Automatically created passwords can be – meaningfully - downloaded through the web service.

Login names

It is usually unwise to overwrite login names as the users may have logged into P247 and changed their login names.

P247 can automatically create login names if no such field is passed to it when creating a user record. P247 uses the surname and appends a few characters to make the name unique. Login names must be globally unique in P247. In an attempt to create two users where one of them has a login name violating the unique constraint, both accounts don't get created. The entire transaction fails.

This could theoretically (though unlikely) also happen when auto-creating login names. In such a case the upload ought to be reattempted.

Login names have significance in the forgot password process, as they may be used as a key to reset an account's password.

Email addresses

It is usually unwise to overwrite email addresses as the users may have logged into P247

and changed their email address.

Login names have significance in the forgot password process, as they may be used as a key to reset an account's password.

There is no unique constraint on email addresses, which causes the forgot password process to fail if there are duplicate emails in the database. In such cases only the login name as a key will work.

The forgot password process, if uniqueness issues don't spoil the show, can be used as a password distribution process. Note that if using this system as the password distribution process, no sensitive data should be stored in any user account before the user has accessed his account.

Creation vs. Modification

If a user record is present in an UploadRequest (PutList), the corresponding record from the database is retrieved or created, depending on the existence of such a record. Then all the fields present in the uploaded record are copied into the database object and the object is finally stored.

Fields not present in the upload remain untouched.

If a record is newly created without a login name field, then a login name is generated by the system.

If a password is not present for a newly created record, then a password is created. Clear text passwords can be reported through the web service or through the web interface.

First Time Users Email

The UserUploadRequest class has a property named FirstTimeUsersEmail, which is an array of Strings.

Each String in the array represents a line in an email that get sent to any newly created user, if an email address is also given.¹ P247 inspects the email body and carries out the following variable replacements:

- 1) \$loginname (as sent in the request or as generated by Pay24-7)
- 2) \$password (as sent in the request or as generated by Pay24-7)
- 3) \$title (as sent in the request)
- 4) \$forename (as sent in the request)
- 5) \$surname (as sent in the request)
- 6) \$sitename (editable by the sitemanager when logged into Pay24-7 under Manage->Site data)
- 7) \$sitedescription (editable by the sitemanager when logged into Pay24-7 under Manage->Site data)

Please note that these variable names are case sensitive.

Further than these replacements, Pay24-7 can manipulate other parts of the email than

¹ The reason the email template needs to be given as an array of String objects is that SOAP and the underlying XML do not preserve linebreaks in a useful way.

the mere email body: Subject, FromAddress and Signature. This can be achieved by sending the following instructions:

- 1) \$Set PlatformName = my trial site for SOAP testing
- 2) \$Set Subject = welcome new user
- 3) \$Set FromAddress = payment-gateway@example-company.net

We give an example below.

Please note that emails get created only for newly created user accounts. A formerly “deleted” i.e. deactivated, and then “created” i.e. reactivated user will not receive an email. The identification of users happens solely through the external identifier, i.e. the identifier used to create, delete and then recreate the user. This identifier is usually the identifier from the client's information management system (IMS).

6.4 Examples

Downloading lists of user records

A bit of coding for the creation of a user account could look similar to the following (again VB.Net)

```
Dim prx As New P247WS.localhost.PubMethods ' proxy class
Dim req As New P247WS.localhost.UserDownloadRequest
req.OrgIdentifier = "myOrganisation"
req.Username = "myUserName"
req.Password = "verySecret"
Dim resp As P247WS.localhost.UserHandlingReceipt
resp = prx.handleUserDownload(req)
```

The object resp contains the objects as described above in the UML diagrams, most notably a property “UserDump” which is an array of UserVO objects. Note that the user accessing the web service needs sufficient privileges to be granted access to the user downloading facility. The XML that gets generated by the request looks like the following:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handleUserDownload xmlns="http://www.pay24-
7.com/P247WS/PubMethods">
      <_req>
        <OrgIdentifier>
          myOrganisation</OrgIdentifier>
        <Password>verySecret</Password>
        <Username>myUserName</Username>
      </_req>
    </handleUserDownload>
  </soap:Body>
</soap:Envelope>
```

And the response looks similar to this:

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

6 User Management

```
<handleUserDownloadResponse
  xmlns="http://www.pay24-7.com/P247WS/PubMethods">
  <handleUserDownloadResult>
    <UserDump>
      <UserVO>
        <Identifier>1</Identifier>
        <Password>Dtbn2J</Password>
        <LastName>1</LastName>
        <LoginName>1.7gTua</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>2</Identifier>
        <Password>geHnTS</Password>
        <LastName>2</LastName>
        <LoginName>2.qLzqk</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>3</Identifier>
        <Password>Bga57A</Password>
        <LastName>3</LastName>
        <LoginName>3.KsqQx</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>4</Identifier>
        <Password>TxT95k</Password>
        <LastName>4</LastName>
        <LoginName>4.nUBjf</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>5</Identifier>
        <Password>5Z2q_j</Password>
        <LastName>5</LastName>
        <LoginName>5.bbLUS</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>6</Identifier>
        <Password>KkPLUy</Password>
        <LastName>6</LastName>
        <LoginName>6.XM5hf</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>7</Identifier>
        <Password>LMHGb8</Password>
        <LastName>7</LastName>
        <LoginName>7.jqkFy</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>8</Identifier>
        <Password>EgYHC9</Password>
        <LastName>8</LastName>
        <LoginName>8.roDgV</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>9</Identifier>
        <Password>UDCMWo</Password>
        <LastName>9</LastName>
        <LoginName>9.4mJZt</LoginName>
      </UserVO>
      <UserVO>
        <Identifier>10</Identifier>
        <Password>PFcX7C</Password>
        <LastName>10</LastName>
        <LoginName>10.49WG2</LoginName>
      </UserVO>
    </UserDump>
  </handleUserDownloadResult>
</handleUserDownloadResponse>
```

```

        </UserVO>
    </UserDump>
    <SuccessState>0</SuccessState>
    <VerboseSuccessState>SUCCESS</VerboseSuccessState>
    <UsersActive>0</UsersActive>
    <UsersAdded>0</UsersAdded>
    <UsersDeactivated>0</UsersDeactivated>
    <UsersInExport>10</UsersInExport>
    <UsersModified>0</UsersModified>
    </handleUserDownloadResult>
</handleUserDownloadResponse>
</soap:Body>
</soap:Envelope>

```

Note that this example was created with an automated test and increasing integers were used for the SurName fields during the creation of these users. Loginnames and passwords were autocreated. It is not always necessary to download the entire set of users. An array of users may be specified by using code similar to the following:

```

Dim prx As New P247WS.localhost.PubMethods ' proxy class
Dim req As New P247WS.localhost.UserUploadRequest
req.OrgIdentifier = "myOrg"
req.Username = "myUserName"
req.Password = "verySecret"
Dim uvo As UserVO
Dim getUsers(3) As UserVO ' array of four UserVOs to download
For i As Int16 = 1 To 4
    uvo = New UserVO
    uvo.Identifier = i.ToString
    getUsers(i - 1) = uvo
Next
req.PutUsers = getUsers
Dim resp As P247WS.localhost.UserHandlingReceipt
resp = prx.handleUserDownload(req)

```

Request and response XML streams look as the following:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handleUserDownload xmlns="http://www.pay24-
7.com/P247WS/PubMethods">
      <_req xsi:type="UserUploadRequest">
        <OrgIdentifier>
          myOrg</OrgIdentifier>
        <Password>verySecret</Password>
        <Username>myUserName</Username>
        <PutUsers>
          <UserVO>
            <Identifier>1</Identifier>
          </UserVO>
          <UserVO>
            <Identifier>2</Identifier>
          </UserVO>
          <UserVO>
            <Identifier>3</Identifier>
          </UserVO>
          <UserVO>
            <Identifier>4</Identifier>
          </UserVO>
        </PutUsers>
      </_req>
    </handleUserDownload>
  </soap:Body>
</soap:Envelope>

```

6 User Management

```
        </_req>
      </handleUserDownload>
    </soap:Body>
  </soap:Envelope>
```

. . and the corresponding response is

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handleUserDownloadResponse
      xmlns="http://www.pay24-7.com/P247WS/PubMethods">
      <handleUserDownloadResult>
        <UserDump>
          <UserVO>
            <Identifier>1</Identifier>
            <Password>drUYCu</Password>
            <LastName>1</LastName>
            <LoginName>1.SAxZE</LoginName>
          </UserVO>
          <UserVO>
            <Identifier>2</Identifier>
            <Password>7RiDnh</Password>
            <LastName>2</LastName>
            <LoginName>2.QJGtM</LoginName>
          </UserVO>
          <UserVO>
            <Identifier>3</Identifier>
            <Password>p5VxZf</Password>
            <LastName>3</LastName>
            <LoginName>3.9Y246</LoginName>
          </UserVO>
          <UserVO>
            <Identifier>4</Identifier>
            <Password>7o433S</Password>
            <LastName>4</LastName>
            <LoginName>4.7uPAH</LoginName>
          </UserVO>
        </UserDump>
        <SuccessState>0</SuccessState>
        <VerboseSuccessState>SUCCESS</VerboseSuccessState>
        <UsersActive>0</UsersActive>
        <UsersAdded>0</UsersAdded>
        <UsersDeactivated>0</UsersDeactivated>
        <UsersInExport>4</UsersInExport>
        <UsersModified>0</UsersModified>
      </handleUserDownloadResult>
    </handleUserDownloadResponse>
  </soap:Body>
</soap:Envelope>
```

this time only containing the requested four user records.

Modifying User Records

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handleUserUpload xmlns="http://www.pay24-7.com/P247WS/PubMethods">
      <_req>
        <OrgIdentifier>
```

6 User Management

```
        myOrg</OrgIdentifier>
        <Password>verySecret</Password>
        <Username>myUserName</Username>
        <PutUsers>
            <UserVO>
                <Title>Mr.</Title>
                <DoB>1970-01-01</DoB>
                <Email>al@lea.al.banthien.net</Email>
                <FirstName>Alexander</FirstName>
                <Identifier>
                    8af6a435-e574-4059-9570-
f1c7398d92c8</Identifier>
                <LastName>Banthien</LastName>
                <LastLogin>06.06.2005 21:39:39</LastLogin>
            </UserVO>
        </PutUsers>
    </_req>
</handleUserUpload>
</soap:Body>
</soap:Envelope>
```

with response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Body>
        <handleUserUploadResponse
            xmlns="http://www.pay24-7.com/P247WS/PubMethods">
            <handleUserUploadResult>
                <SuccessState>0</SuccessState>
                <VerboseSuccessState>SUCCESS</VerboseSuccessState>
                <UsersActive>0</UsersActive>
                <UsersAdded>1</UsersAdded>
                <UsersDeactivated>0</UsersDeactivated>
                <UsersInExport>0</UsersInExport>
                <UsersModified>0</UsersModified>
            </handleUserUploadResult>
        </handleUserUploadResponse>
    </soap:Body>
</soap:Envelope>
```

created a single new user record with 8af6a435-e574-4059-9570-f1c7398d92c8 as identifier. If the very same request (same identifier!) would be sent again the web service would report one modified but 0 added records. The VB.Net code for the above request looked like this:

```
Dim prx As New P247WS.localhost.PubMethods ' proxy class
Dim req As New P247WS.localhost.UserUploadRequest
Dim rec As New P247WS.localhost.UserHandlingReceipt
Dim guid As String = System.Guid.NewGuid.ToString
req.OrgIdentifier = "myOrg"
req.Username = "myUserName"
req.Password = "verySecret"
Dim uvo As New P247WS.localhost.UserVO
uvo.DoB = "1970-01-01"
uvo.Email = "al@lea.al.banthien.net"
uvo.Title = "Mr."
uvo.FirstName = "Alexander"
uvo.LastName = "Banthien"
uvo.Identifier = guid
uvo.LastLogin = Date.Now.ToString
```

6 User Management

```
'uvo.LoginName = "test1ksjhfkdfjhs" ' if loginname not sent, name chosen  
'uvo.Password = "password" ' if password not sent, password generated cleartext  
Dim userList() As UserVO = {uvo}  
req.PutUsers = userList  
  
rec = prx.handleUserUpload(req)
```

Creating a new user and sending a First Time User Email

We give an example in VB.Net:

```
Dim prx As New P247WS.localhost.PubMethods ' proxy class  
prx.Url = "https://www.pay24-7.com/P247WS/PubMethods.asmx"  
Dim rec As P247WS.localhost.UserHandlingReceipt  
Dim req As New P247WS.localhost.UserUploadRequest  
Dim text() As String = New String() { _  
    "Dear $title $forename $surname", _  
    "your username and password are: ", _  
    "$loginname and $password", _  
    "thank you for using our payment system", _  
    "$sitename", _  
    "$Set PlatformName = my trial site for SOAP testing", _  
    "$Set Subject = welcome new user", _  
    "$Set FromAddress = payment-gateway@example-company.net"}  
req.FirstTimeUsersEmail = text  
  
Dim uvo As New P247WS.localhost.UserVO  
uvo.Title = "Mrs."  
uvo.FirstName = "First"  
uvo.LastName = "Last"  
uvo.Email = "al@lea.al.banthien.net"  
uvo.Identifier = userId  
req.PutUsers = New P247WS.localhost.UserVO() {uvo}  
  
req.OrgIdentifier = orgIdentifier  
req.Username = "userWithSufficientPrivs"  
req.Password = "secret"  
prx.handleUserUpload(req)
```

This results in the following SOAP XML to be created:

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
    <soap:Body>  
        <handleUserUpload xmlns="http://www.pay24-7.com/P247WS/PubMethods">  
            <_req>  
                <OrgIdentifier>  
                    e6b2dd8a-4604-4548-a6ae-60346520add8</OrgIdentifier>  
                <Password>repldown</Password>  
                <Username>6bd5c272-ce97-4eda-8639-5b08be43257d</Username>  
                <PutUsers>  
                    <UserVO>  
                        <Title>Mrs.</Title>  
                        <Email>dearuser@email.com</Email>  
                        <FirstName>First</FirstName>  
                        <Identifier>  
                            727e233c-75ea-445d-bdb3-1420010f4b4c</Identifier>  
                        <LastName>Last</LastName>
```

6 User Management

```
        </UserVO>
    </PutUsers>
    <FirstTimeUsersEmail>
        <string>Dear $title $forename $surname,</string>
        <string>your username and password are:</string>
        <string>$loginname and $password</string>
        <string>thank you for using our payment system</string>
        <string>$sitename</string>
        <string>$Set PlatformName = my trial site for SOAP
            testing</string>
        <string>$Set Subject = welcome new user</string>
        <string>$Set FromAddress =
            payment-gateway@example-company.net</string>
    </FirstTimeUsersEmail>
</_req>
</handleUserUpload>
</soap:Body>
</soap:Envelope>
```

And the server responds with the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handleUserUploadResponse
      xmlns="http://www.pay24-7.com/P247WS/PubMethods">
      <handleUserUploadResult>
        <SuccessState>0</SuccessState>
        <VerboseSuccessState>SUCCESS</VerboseSuccessState>
        <UsersActive>0</UsersActive>
        <UsersAdded>1</UsersAdded>
        <UsersDeactivated>0</UsersDeactivated>
        <UsersInExport>0</UsersInExport>
        <UsersModified>0</UsersModified>
      </handleUserUploadResult>
    </handleUserUploadResponse>
  </soap:Body>
</soap:Envelope>
```

...and also sends an email from `payment-gateway@example-company.net` to `dearuser@email.com` with subject `welcome new user` the following content:

```
Dear Mrs. First Last,
your username and password are:
Last.u_7pM and r8SPMQ
thank you for using our payment system
SOAP Testing
```

```
=====
Email sent to you by my trial site for SOAP testing
Email sent on 24.08.2005 - 17:25:30
```

Deleting users

The following code deletes a single user record (with identifier 10). Several identifiers may clearly be specified.

6 User Management

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handleUserUpload xmlns="http://www.pay24-7.com/P247WS/PubMethods">
      <_req>
        <OrgIdentifier>
          myOrg</OrgIdentifier>
        <Password>verySecret</Password>
        <Username>myUserName</Username>
        <DeleteUsers>
          <UserVO>
            <Identifier>10</Identifier>
          </UserVO>
        </DeleteUsers>
      </_req>
    </handleUserUpload>
  </soap:Body>
</soap:Envelope>
```

The response was

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <handleUserUploadResponse
      xmlns="http://www.pay24-7.com/P247WS/PubMethods">
      <handleUserUploadResult>
        <SuccessState>0</SuccessState>
        <VerboseSuccessState>SUCCESS</VerboseSuccessState>
        <UsersActive>0</UsersActive>
        <UsersAdded>0</UsersAdded>
        <UsersDeactivated>1</UsersDeactivated>
        <UsersInExport>0</UsersInExport>
        <UsersModified>0</UsersModified>
      </handleUserUploadResult>
    </handleUserUploadResponse>
  </soap:Body>
</soap:Envelope>
```

and the VB.Net code for this request had been:

```
Dim prx As New P247WS.localhost.PubMethods ' proxy class
Dim req As New P247WS.localhost.UserUploadRequest
req.OrgIdentifier = "myOrg"
req.Username = "myUserName"
req.Password = "verySecret"
Dim deleteUvo As New UserVO
deleteUvo.Identifier = 10
req.DeleteUsers = New UserVO() {deleteUvo}

Dim resp As P247WS.localhost.UserHandlingReceipt
resp = prx.handleUserUpload(req)
```

7 User Messages

For the purpose of e.g. account balance reporting, or other purposes of sending messages to users the methods handle(Simple)UserMessageRequest allow for messages to be sent onto the home page of each site user. Each message will be truncated to 200 characters. The "simple" version of the methods allows a message to be stored against a single user,

7 User Messages

identified by his identifier. The complex version allows the simultaneous upload of several messages against several users. Messages and user ids are sent in two separate arrays, please make sure the order in both arrays is the same.

Calls of this method are potentially very bandwidth consuming. Pay24-7/ParentPay reserves the right to limit access to this function. Please use it sensibly by sending at most one message update per user and day and aggregating as many updates into one complex call as is reasonably possible.

Terms of Use

© 2006 ParentPay Limited and affiliated companies. All Rights Reserved.

ParentPay™, ParentPay™ for Schools, ParentPay™ mealScene, ParentPay™ Dinner Money, ParentPay™ eDm² and Pay24-7™ are all trademarks of ParentPay Limited

This document contains unpublished, confidential and proprietary information of ParentPay Limited that is protected by copyright and other intellectual property laws. All other trademarks, service marks, and trade names referenced herein are the property of their respective owners.

ParentPay Limited software and its attendant documentation are subject to and made available only pursuant to the terms of a User Agreement with ParentPay Limited, and may be used or copied only in accordance with the terms of that agreement.

ParentPay Limited takes reasonable measures to ensure the quality of the data and other information produced by ParentPay Limited that is made available herein. However, these materials may contain technical inaccuracies or typographical errors, and are not guaranteed to be error-free. Information may be changed or updated without notice. ParentPay Limited has no obligation to update the information in this material based on changes to its products or services or those of third parties. ParentPay Limited may also make improvements or changes to the products or services described in this information at any time without notice.