project portfolio office

PPO User Management API

project portfolio office

Contents

1 Introduction

To facilitate programmatic user management, PPO provides an API in the form of a SOAP web service. The API provides for the following:

- Retrieving details for one or all users
- Adding a user
- Updating various user attributes
- Resetting a user's password

This document serves as a technical guide for software developers on how to use the functionality that is available in the API.

1.1 Service end-point

The service end-point to the API is available at the following URL:

https://www.ppolive.com/{yourinstancename}/webservices/Users.asmx

Replace the {yourinstancename} tag with your actual instance name. Note that the API end-point is only available over HTTPS.

1.2 Session state and cookie container

For PPO to maintain session state, the calling client must support cookies. With browsers, this is built-in and comes enabled by default. From third-party client tools and custom developed applications, some type of cookie container needs to be provided.

Once authenticated, there is no need to do so again for as long as the session is active. PPO's user sessions time out after 60 minutes of inactivity. We will show you later in this document how to do this in both Visual Studio and Java.

1.3 Authentication and LoginExplicit

Authentication is simply done by calling the **LoginExplicit** web method with the correct credentials of a PPO user account. After authenticating you can make additional web service requests as long as you maintain the session in a cookie container as described above.

If you make web service requests without first calling LoginExplicit, you will get an authentication failure exception. This exception is explained in the PPO exceptions section.

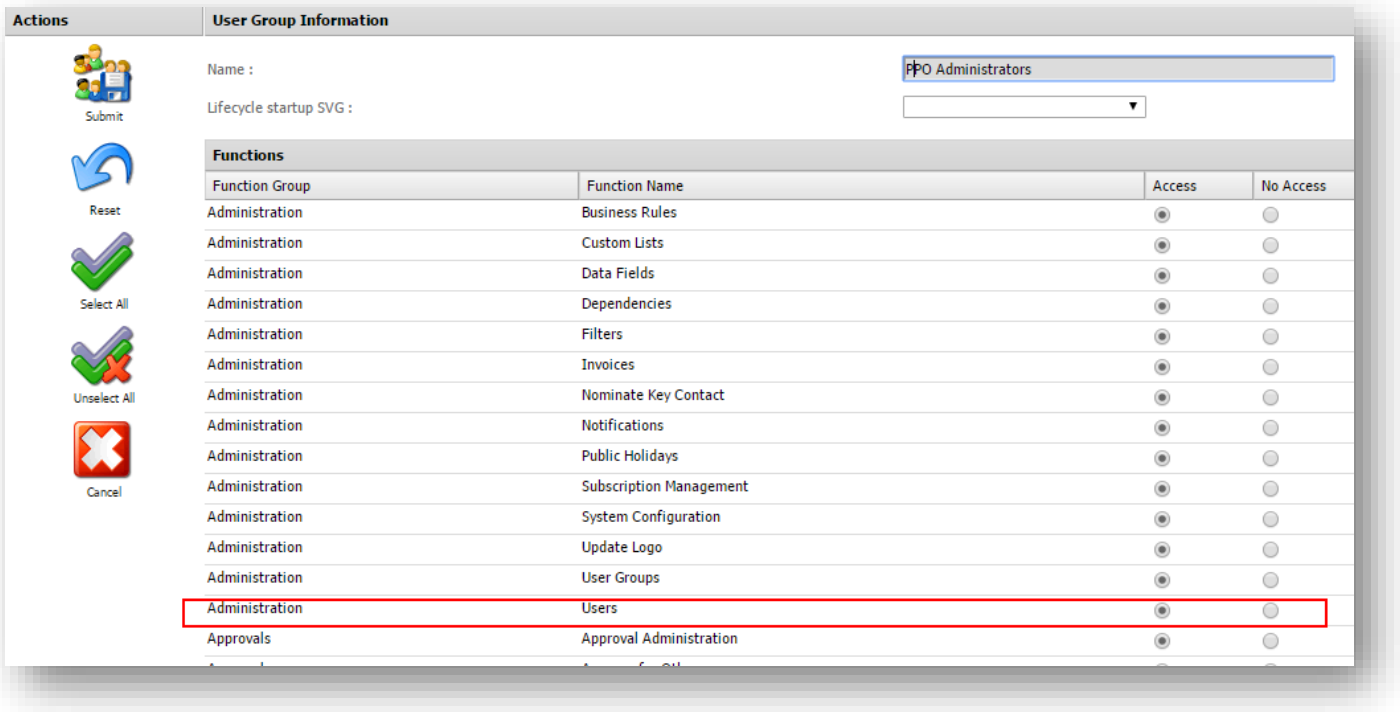
All code samples provided in the rest of this document assume that the authentication has already been done and that a cookie container exists.

To abandon your session, the Logout web method should be called.

It is always good practice to ensure that your credentials are encrypted and secure.

1.4 Authorisation

The API utilises our existing logical access control configured from the UI under Administration | User Groups.



In order to call any of the methods of this API, the user that is used must have access to the Administration / Users function as shown in the screenshot above.

1.5 Fair usage

Please refer to our Terms and Conditions.

1.6 Exception handling

Refer to the Exceptions section for each web method or the General Exceptions section at the end of the document.

1.7 API Response

Besides the Retrieve All and Reset User Password web methods, most web methods in the API will have a similar response represented in an XML format as shown below.

```
<user key="508">
  <username>luke.skywalker</username>
  <usergroupkey>1</usergroupkey>
  <usergroupname>PPO Administrators</usergroupname>
  <employeekey>3829</employeekey>
  <employeename>Luke Skywalker</employeename>
  <active>true</active>
  <lastlogin></lastlogin>
</user>
```

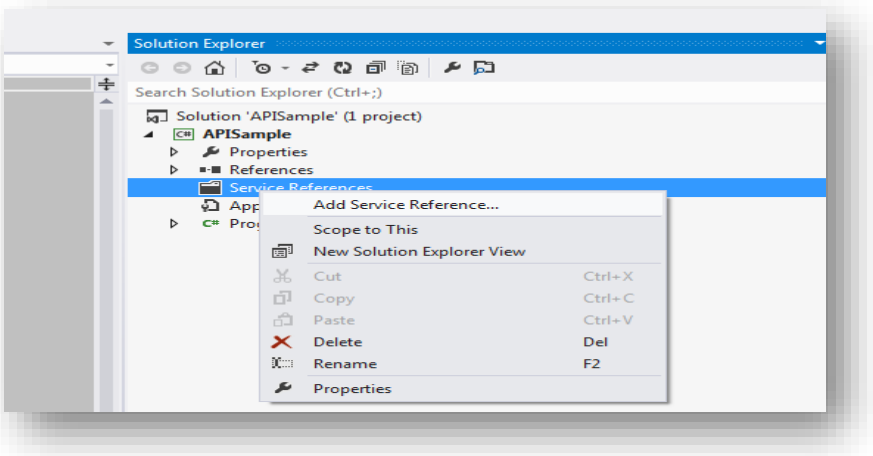
2 Getting started with Visual Studio and C#

Before the SOAP web service can be consumed by your client application, a web reference is required. Visual Studio will generate the web service proxy code for you to use.

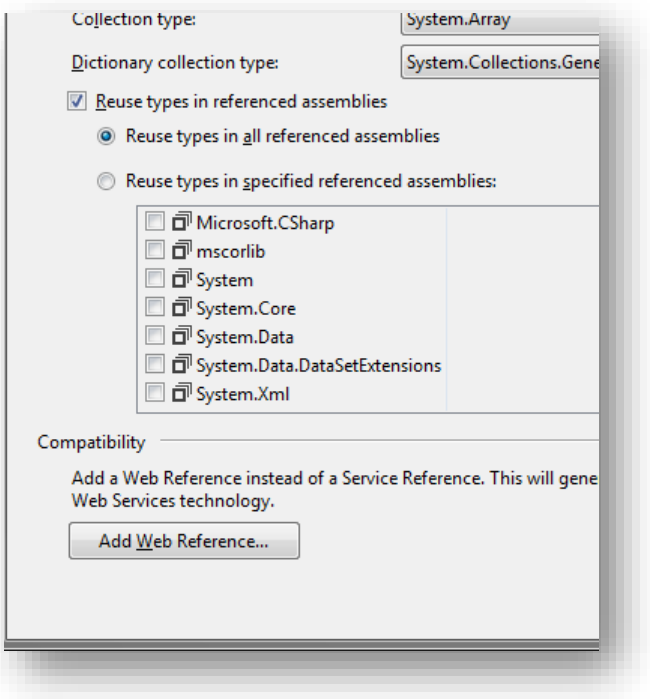
2.1 Adding the Web Reference

To setup a reference to the service end-point, follow these steps:

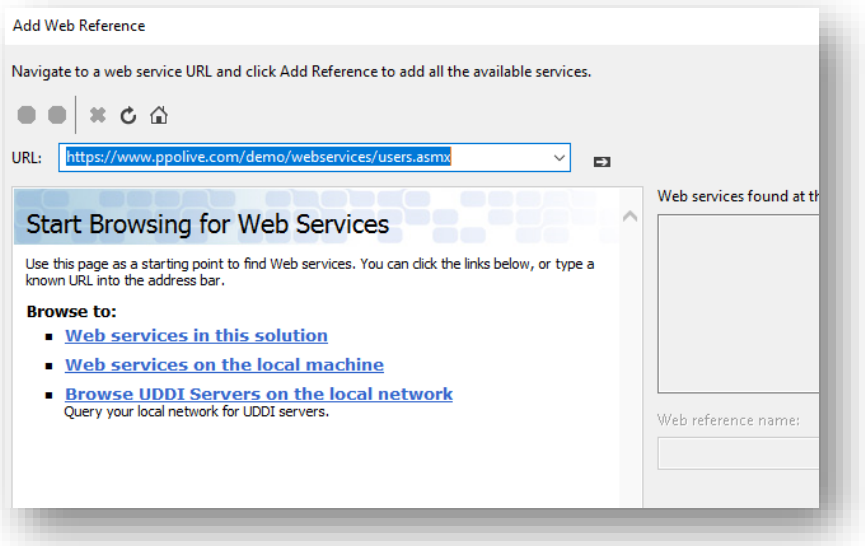
Add Service Reference, and choose **Advanced**.



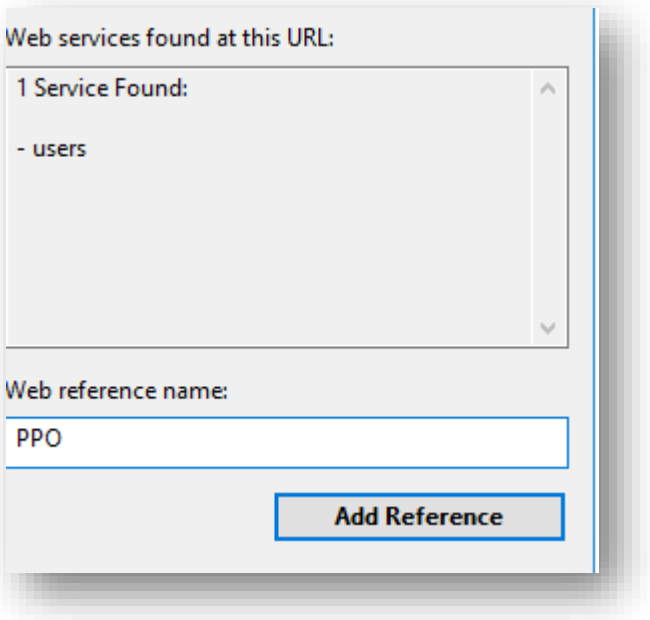
1. **Add Web Reference**



2. Enter the service **URL** and click the arrow button.



3. Enter a **Web reference name**, click **Add Reference**



At this point, Visual Studio will generate client proxy code that is then used for calling web methods as well as setup the configuration for the web reference in your configuration file.

2.2 Authentication through code in C#

The following sample shows how to authenticate and maintain session state using a cookie container:

```
namespace APISample
{
    class Program
    {
        static void Main(string[] args)
        {
            PPO.UserWebService api = new PPO.UserWebService
            {
                CookieContainer = new System.Net.CookieContainer()
            };
            var userId = api.LogInExplicit("api.user", "pwd123");
        }
    }
}
```

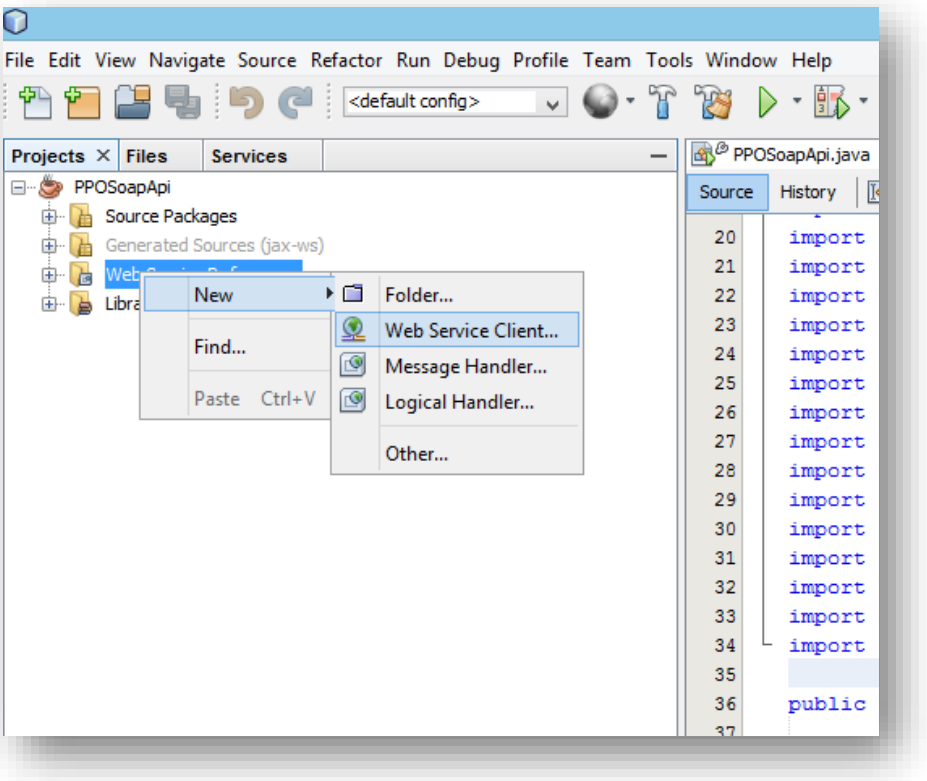
3 Getting started with the NetBeans IDE and Java

Like Visual Studio, NetBeans will generate client proxy code that will enable you to consume PPO's API web methods.

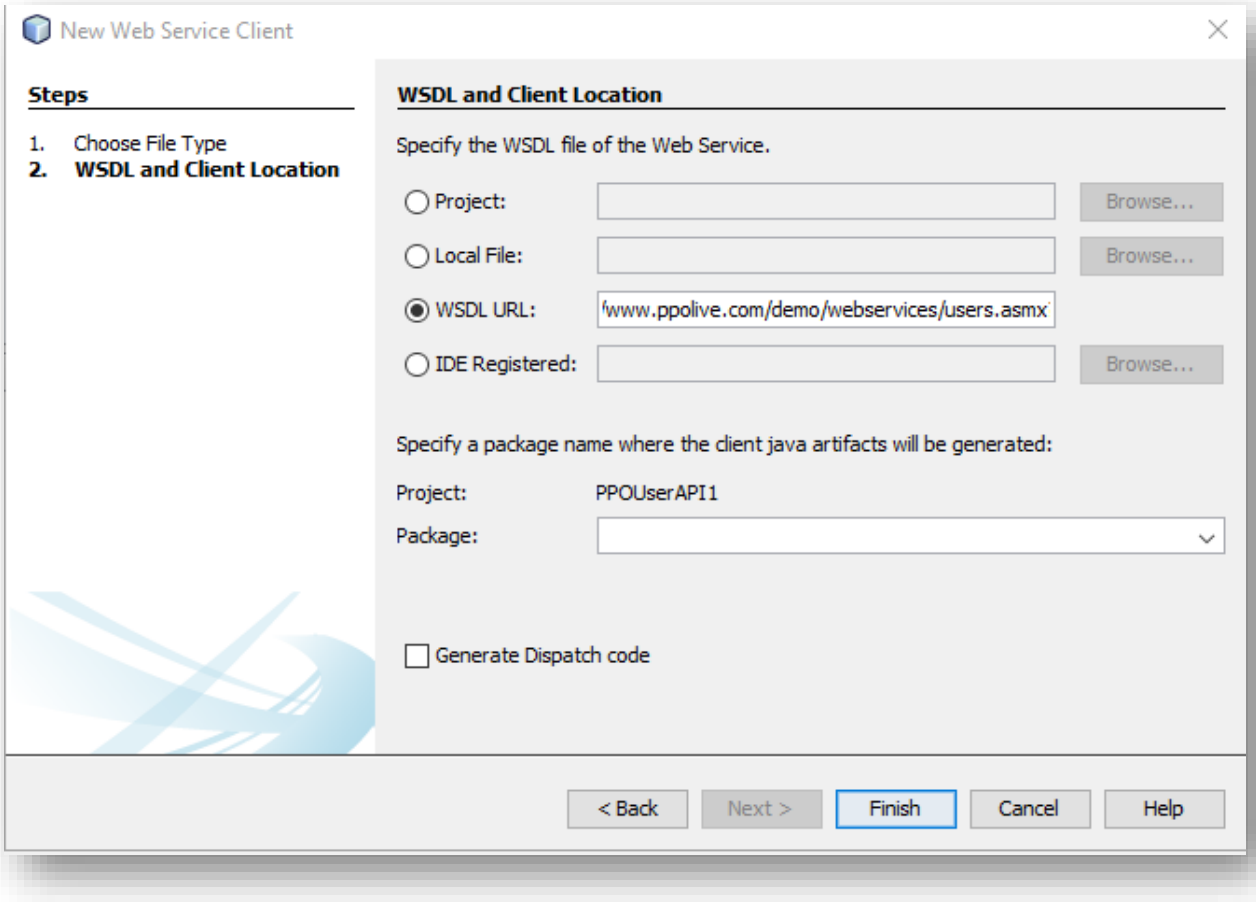
3.1 Web Reference

To setup a reference to the service end-point, follow these steps:

1. Right click on **Web Service References** then **New** and then select **Web Service Client**.



2. Choose **WSDL URL** then enter the service **URL** and click **Finish**


New Web Service Client
Steps
1. Choose File Type
2. WSDL and Client Location
WSDL and Client Location
Specify the WSDL file of the Web Service.
Project:
Local File:
WSDL URL:
www.ppolive.com/demo/webservices/users.asmx
IDE Registered:
Browse...
Specify a package name where the client java artifacts will be generated:
Project:
PPOUserAPI1
Package:
Generate Dispatch code
< Back
Next >
Finish
Cancel
Help

3.2 Authentication

The following code sample shows how to authenticate and set the session cookie in the request header:

```
public static void main(String[] args) throws MalformedURLException
    {
        String url = "https://www.ppolive.com/demo/webservices/users.asmx?wsdl";
        UsersWebService service = new UsersWebService(new URL(url));
        UsersWebServiceSoap api = service.getUsersWebServiceSoap();
        BindingProvider bindingProvider = getBindingProvider(api);

        Integer userKey = api.logInExplicit("project.manager", "password123");

        setCookieHeaders(bindingProvider, getCookieHeaders(bindingProvider));
    }

private static BindingProvider getBindingProvider(IntegrationSoap api)
    {
        BindingProvider bindingProvider = (BindingProvider) api;
        bindingProvider.getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
        return bindingProvider;
    }

private static String getCookieHeaders(BindingProvider bindingProvider)
    {
        return ((Map<String, List<String>>)bindingProvider
                        .getResponseContext()
                        .get(MessageContext.HTTP_RESPONSE_HEADERS))
                        .get("Set-Cookie").toString();
    }

private static void setCookieHeaders(BindingProvider bindingProvider, String cookieHeaders)
    {
        bindingProvider
                .getRequestContext()
                .put(
                        MessageContext.HTTP_REQUEST_HEADERS,
                        Collections.singletonMap("Cookie",
                        Collections.singletonList(cookieHeaders)));
    }
```

4 PPO User API Web Methods

This section of the document will describe the use of each web method on the user API, the parameters required by each web method and lastly the types of exceptions that could occur when executing each web method.

4.1 RetrieveUserByKey

Retrieves a user by its corresponding user key

Parameters

Name	Data Type	Description
userKey	Integer	Unique user identifier e.g. UserKey=4

Returns

This method returns a user object in an XML format as described in 1.7

Example

Java Code Sample

```
String user = api.RetrieveByUserKey(508);
```

C# Code Sample

```
var user = api.RetrieveByUserKey(508);
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.NoAccessException
- PPO.InvalidUserKeyException
- System.ArgumentException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.2 RetrieveUserByUserName

Retrieves a user by their unique username.

Parameters

Name	Data Type	Description
username	String	Unique name of a user e.g. 'john.doe'

Returns

This method returns a user object in an XML format as described in 1.7

Example

Java Code Sample

```
String user = api.RetrieveUserByUsername("luke.skywalker");
```

C# Code Sample

```
var user = api.RetrieveUserByUsername("luke.skywalker");
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.NoAccessException
- System.ArgumentException
- PPO.MaximumStringLengthException
- PPO.MinimumStringLengthException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.3 RetrieveAll

Returns a list of all users.

Parameters

This web method requires no parameters.

Returns

This method returns the details of all users in XML format as shown under the Response section below.

Example

Java Code Sample

```
String user = api.RetrieveAll();
```

C# Code Sample

```
var user = api.RetrieveAll();
```

Response

```
<users>
  <<user key="4">
    <username>james.bond</username>
    <usergroupkey>9</usergroupkey>
    <usergroupname>Team Members (Clients)</usergroupname>
    <employeekey>112</employeekey>
    <employeename>James Bond</employeename>
    <active>false</active>
    <lastlogin>2016-12-06 09:41:40</lastlogin>
  </user>
  <user key="11">
    <username>jane.bond</username>
    <usergroupkey>1</usergroupkey>
    <usergroupname>PPO Administrators</usergroupname>
    <employeekey>118</employeekey>
    <employeename>Jane Bond</employeename>
    <active>true</active>
    <lastlogin></lastlogin>
  </user>
  <user key="6">
    <username>john.wick</username>
    <usergroupkey>1</usergroupkey>
    <usergroupname>PPO Administrators</usergroupname>
    <employeekey>16136</employeekey>
    <employeename>John Wick</employeename>
    <active>false</active>
    <lastlogin>2016-09-14 03:03:30</lastlogin>
```

```
  </user>
  <user key="13">
    <username>mary.bond</username>
    <usergroupkey>9</usergroupkey>
    <usergroupname>Team Members (Clients)</usergroupname>
    <employeekey>114</employeekey>
    <employeename>Mary Bond</employeename>
    <active>true</active>
    <lastlogin></lastlogin>
  </user>
</users>
```

Exceptions

- PPO.NoAccessException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.4 AddUser

Creates a new user.

Parameters

Name	Data Type	Description
username	String	A unique name of the new user being created.
userGroupKey	Short	The key that represents which user group the new user belongs to
employeeKey	Integer	The resource key that the created user will be associated with.

Returns

This method returns a user object in an XML format as described in 1.7

Example

Java Code Sample

```
String user = api.AddUser("luke.skywalker", 1, 3829);
```

C# Code Sample

```
var user = api.AddUser("luke.skywalker", 1, 3829);
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.NoAccessException
- PPO.DuplicateUserNameException
- PPO.UserCountExceededException
- PPO.MinimumStringLengthException
- PPO.MaximumStringLengthException
- PPO.EmployeeHasNoEmailAddressException
- System.ArgumentException
- PPO.UnableToRetrieveRecordException
- PPO.InvalidUserKeyException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.5 UpdateUsername

Updates a user's username by their corresponding user key.

Parameters

Name	Data Type	Description
userKey	Integer	Unique user identifier e.g. UserKey=4
userName	String	A unique name associated with the user

Returns

This method returns a user object in an XML format as described in 1.7

Example

Java Code Sample

```
String user = api.UpdateUserName(508, "annikan.skywalker");
```

C# Code Sample

```
var user = api.UpdateUserName(508, "annikan.skywalker");
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.DuplicateUserNameException

- PPO.NoAccessException
- PPO.InvalidUserKeyException
- PPO.MinimumStringLengthException
- PPO.MaximumStringLengthException
- System.ArgumentException
- PPO.CannotChangeSystemAdministratorException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.6 UpdateUserEmployee

Updates the resource that the user is associated with.

Parameters

Name	Data Type	Description
userKey	Integer	A unique user identifier e.g. UserKey=4
employeeKey	Integer	The resource key that the user will be associated with.

Returns

This method returns a user object in an XML format as described in 1.7

Example

Java Code Sample

```
String user = api.UpdateUserEmployee(508, 3830);
```

C# Code Sample

```
var user = api.UpdateUserEmployee(508, 3830);
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.NoAccessException
- PPO.EmployeeHasNoEmailAddressException
- System.ArgumentException
- PPO.UnableToRetrieveRecordException
- PPO.InvalidUserKeyException
- PPO.CannotChangeSystemAdministratorException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.7 UpdateUserUserGroup

Updates the user group that the user belongs to.

Parameters

Name	Data Type	Description
userKey	Integer	A unique user identifier e.g. UserKey=4
userGroupKey	Short	The key that represents which user group the user belongs to

Returns

This method returns a user object in an XML format as described in 1.7

Example

Java Code Sample

```
String user = api.UpdateUserUserGroup(508, (Short)21);
```

C# Code Sample

```
var user = api.UpdateUserUserGroup(508, 21);
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.NoAccessException
- System.ArgumentException
- PPO.InvalidUserKeyException
- PPO.CannotChangeSystemAdministratorException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.8 UpdateUserActiveStatus

Updates the active status of the specified user.

Parameters

Name	Data Type	Description

userKey	Integer	A unique user identifier e.g. UserKey=4
active	Boolean	A field that specifies if the created user is in/active.

Returns
This method returns a user object in an XML format as described in 1.7

Example
Java Code Sample

```
String user = api.UpdateUserActiveStatus(508, false);
```

C# Code Sample

```
var user = api.UpdateUserActiveStatus(508, false);
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.NoAccessException
- PPO.UserCountExceededException
- System.ArgumentException
- PPO.InvalidUserKeyException
- PPO.CannotChangeSystemAdministratorException
- PPO.CannotSetKeyContactInactiveException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

4.9 ResetUserPassword

Resets a user's password to a system generated password and mails the password to the e-mail address of the resource associated with the user.

Parameters

Name	Data Type	Description
userKey	Integer	A unique user identifier e.g. UserKey=4

Returns
This web method has no return type (void).

Example

Java Code Sample

```
api.ResetUserPassword(508);
```

C# Code Sample

```
api.ResetUserPassword(508);
```

Exceptions

- PPO.ObjectDoesNotExistException
- PPO.NoAccessException
- System.ArgumentException
- PPO.InvalidUserKeyException
- PPO.CannotChangeSystemAdministratorException

Refer to section 0 at the end of the document for a detailed explanation of each of the exceptions.

5 PPO Exceptions

5.1 PPO.AuthenticationFailureException

PPO.AuthenticationFailureException: User not authenticated

Cause
The user made web service requests without being authenticated, or authenticated but did not maintain session state in a cookie container.

Solution
Authentication is simply done by calling the **LoginExplicit** web method with correct credentials of a PPO user account. Ensure that you are maintaining session state as described in the Getting Started section.

5.2 PPO.NoAccessException

PPO.NoAccessException: Members of this User Group, do not have access to this resource.

Cause
The user that is making the web service call should have access to the User Administration feature as described in the Authorization section.

Solution
Ensure that the user belongs to a user group that has access to administer users as described in the Authorisation section.

5.3 PPO.ObjectDoesNotExistException

PPO.ObjectDoesNotExistException: ObjectDoesNotExistException, Object does not exist.

Cause
The value specified for the user key or user group key parameter represents a user key or user group key that does not exists.

Solution
The client should specify a valid user key or user group key for the user being created/updated. This can be determined by the key on the URL, when editing a user/user group in PPO.

5.4 PPO.UserCountExceededException

PPO.UserCountExceededException: User Count Exceeded Exception.

Cause
You can only add a user if the total number of active users is less than the number of subscriptions that you have signed up for.

Solution
You either need to make some existing users inactive or increase your number of subscriptions. You can do this by accessing subscription management on the administration page.

5.5 PPO.EmployeeHasNoEmailAddressException

PPO.EmployeeHasNoEmailAddressException: Resource Has No Email Address

Cause
When creating a new user, resetting a password and when updating a user's resource record, this exception will be triggered when the user's corresponding resource does not have an email address.

Solution
Ensure that the associated resource record of the user has an email address. The email address can be checked in the resource view page in PPO.

5.6 PPO.CannotSetKeyContactInactiveException

PPO.CannotSetKeyContactInactiveException: User cannot be set to inactive since the user is a current key contact. Nominate another user as key contact first

Cause
PPO ensures that there is always an active user specified as a Key contact. You, therefore, cannot make the current Key contact user inactive.

Solution
Designate another user as the Key Contact through the PPO front-end.

5.7 PPO.DuplicateUserNameException

PPO.DuplicateUserNameException: User name already exists [Username: john.doe]

Cause
The value specified for the user name parameter is the same as that of an existing user. User names must be unique.

Solution
The client should specify a unique username for the user being created.

5.8 PPO.UnableToRetrieveRecordException

PPO.UnableToRetrieveRecordException: Unable to retrieve record from the database.

Cause
This exception can occur when attempting to create/update a user with a resource key that does not exists or updating the resource key for a user to one that does not exist

Solution
Specify a valid resource key for the user being created/updated. This can be determined by the key in the URL, when editing a resource in PPO.

5.9 System.ArgumentException

System.ArgumentException: Cannot convert input to system.int32/system. string

Cause
This will be triggered if the client executes any of the web methods with an invalid input.

Solution
A valid input should be specified. A valid input can be determined by the looking at the data type in the parameters section for each web method.

5.10 PPO.InvalidUserKeyException

PPO.InvalidUserKeyException: Invalid user key

Cause
This exception is only thrown if the UserKey parameter specified as a parmater is 0, which is not a valid user key.

Solution
Provide a valid UserKey parameter.