project portfolio office

PPO SOAP API

project portfolio office

Contents

1 Introduction

To facilitate system to system integration, PPO provides an API (Application Programming Interface) by means of a SOAP web service.

The SOAP (Simple Object Access Protocol) is a messaging protocol that uses XML (Extensible Markup Language) over HTTP (Hypertext Transfer Protocol). SOAP is also platform and language independent and has been supported by numerous vendors and services for means of integration for many years.

PPO's web API consists of a collection of web services that allow for basic operations against PPO's configurable entities.

The purpose of this document is to serve as a guide to software developers for technical purposes.

1.1 Service end-point

The service end-point is available from https://www.ppolive.com/{yourinstancename}/webservices/integration.asmx. Simply substitute your instance name into the URL. Note that the end-point uses HTTPS (only SSL secured session accepted).

1.2 Session state and cookie container

In order for PPO to maintain session state, the calling client has to 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.

This document provides some examples to maintain the session state through code.

1.3 Authentication and LoginExplicit

Authentication is simply done by calling the LogInExplicit web method with correct credentials of a PPO user account. The response from the server will be the user key and also a session

cookie in the header of the response that should be maintained using any form of cookie container. After authenticating you can make additional web service requests.

All code samples of the web methods themselves 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 from the human eye.

1.4 Authorisation

PPO's web methods utilises our existing logical access control configured from the UI under Administration | User Groups.

This document will indicate what the authorisation requirements are for each web method call.

1.5 Entity types and XSD

Updating entities require the entity data to comply with a specific XML schema.

The schema for entity type can be retrieved by using the GetXsd web method with the corresponding entity type passed as a parameter. The response contains the appropriate XSD.

1.6 Custom lists and items

Drop-down lists are called custom lists in PPO and can be maintained by the client's PPO Administrator. Please refer to the Knowledge Base Article for more information.

1.7 Fair usage

Please refer to our Terms and Conditions.

1.8 Exception handling

For detailed information on exceptions, refer to the Exceptions section for each web method or the General Exceptions section at the end of the document.

1.9 General concepts

Retrieving an entity type's key

To find the key for an entity type:

1. In PPO, go to **Administration** and then **Data Fields.**
2. Select the appropriate entity from the drop down.
3. The key will appear in the address bar.

Entity XML

When retrieving entities via the service, the service will respond with XML representing the entity details. The XML will contain a root tag consisting of the entity type name and a key attribute. The following tags will be part of every entity XML:

- 'created' – A date and time showing when the entity was last updated.
- 'userkey' – Shows the user key of the user who updated the entity.
- 'user' – The first and last name of the user who updated the entity.
- 'useremployeekey' – The employee key of the user that updated the entity.
- 'masterkey' – This value is null or could be a key of another entity, which indicates this entity is a history item of the entity with the corresponding key.

After these tags will follow the fields of the entity type with their values and an 'allfields' section which contains all the fields with their actual display values.

Entity list XML

The XML of entity list retrievals (GetEntities, GetEntitiesFiltered, etc.) always has a root tag of the entity type name combined with list, i.e. a list of employee entities will have a root tag such as "employeelist".

The root element also has two attributes named "totalrowcount" and "startindex". The totalrowcount attribute indicates how many records of the specified entity type the XML contains. The startindex attribute is only used for paged retrievals, which is currently not possible through the API.

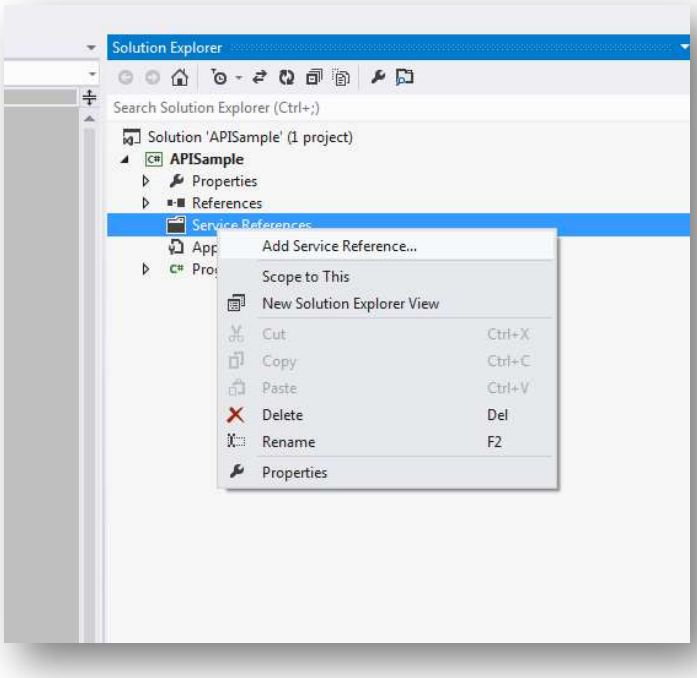
2 Getting started with Visual Studio and C#

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.

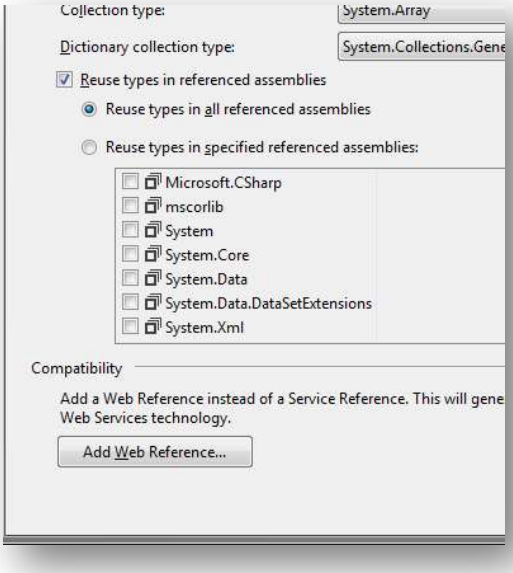
2.1 Adding the Web Reference

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

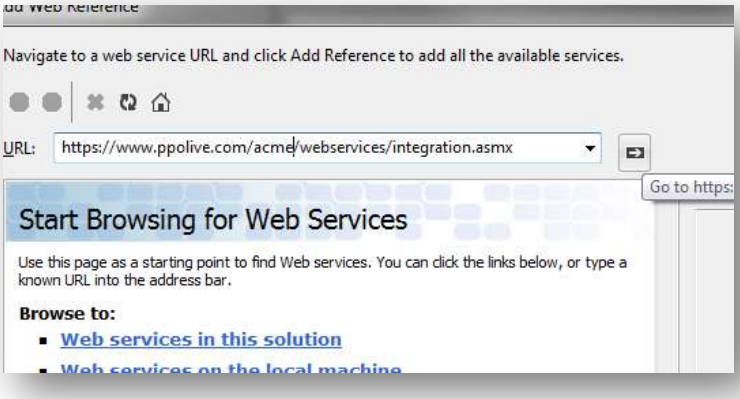
1. **Add Service Reference**, and choose **Advanced**.



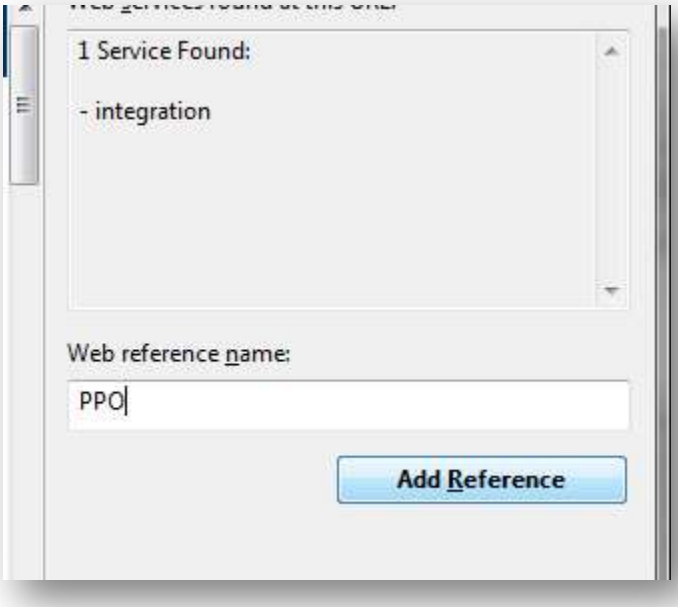
2. **Add Web Reference**



3. Enter the service **URL** and click the arrow bottom



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



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

2.2 Authentication

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

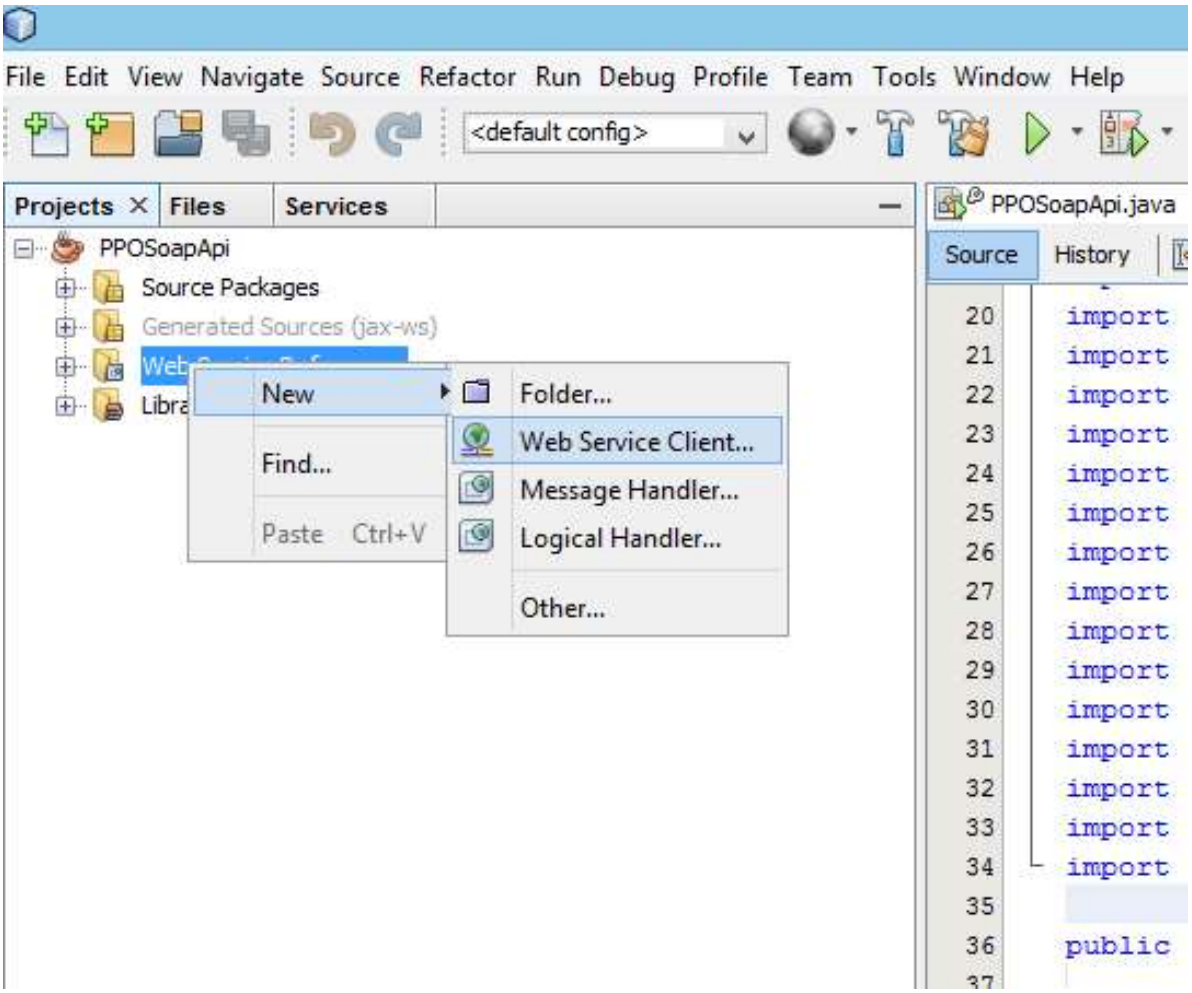
```
namespace APISample
{
    class Program
    {
        static void Main(string[] args)
        {
            PPO.Integration api = new PPO.Integration
            {
                CookieContainer = new System.Net.CookieContainer()
            };
            var userId = api.LogInExplicit("api.user",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

3 Getting started with NetBeans IDE & Java

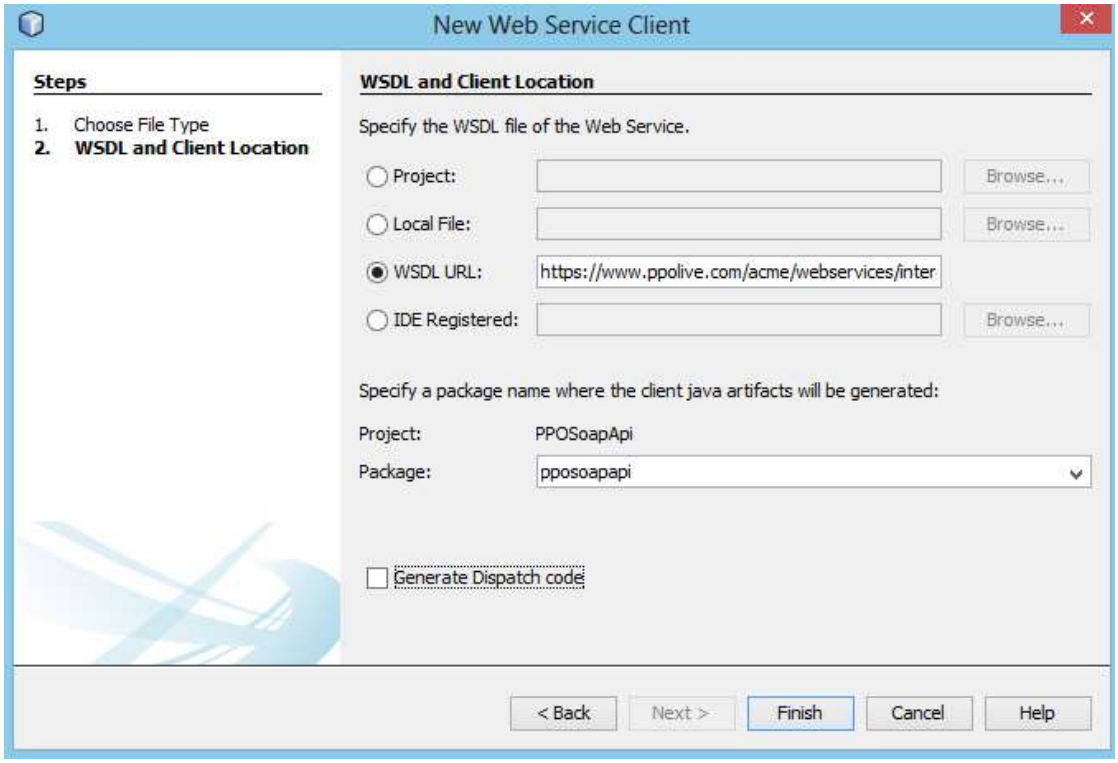
3.1 Web Reference

In order 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**



NOTE: If you encounter the error "undefined simple or complex type 'soapenc:Array'", follow these steps:

- Save WSDL as a file.
- Replace *<s:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>* with *<s:import namespace="http://schemas.xmlsoap.org/soap/encoding/" schemaLocation="http://schemas.xmlsoap.org/soap/encoding/"/>*
- Use the WSDL file to create the web service client instead of the URL.

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
    {
        Integration service = new Integration();
        IntegrationSoap api = service.getIntegrationSoap();
        BindingProvider bindingProvider = (BindingProvider)api;
        bindingProvider.getRequestContext().put(BindingProvider.SESSION_MAINTAIN_PROPERTY, true);
        Integer userKey = api.logInExplicit("project.manager", "password123");
        String cookieHeaders = ((Map<String, List<String>>)bindingProvider
                                .getResponseContext()
                                .get(MessageContext.HTTP_RESPONSE_HEADERS))
                                .get("Set-Cookie").toString();
        bindingProvider
          .getRequestContext()
          .put(MessageContext.HTTP_REQUEST_HEADERS,
              Collections.singletonMap("Cookie", Collections.singletonList(cookieHeaders)));
    }
```

4 PPO API web methods

This section describes the purpose of each web method that our API supports, the logical access required to use them, the parameters they expect, samples of values returned and exceptions that could occur.

4.1 GetEntity

Retrieves XML that contains all the details for the specified PPO entity represented by the entity type and key.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO (refer 1.8.1).
entityKey	Integer	The unique key of the specific entity. (EmployeeKey, TaskKey, etc.)

Returns

XML that represents an entity with all its details.

Example

To retrieve a single employee record.

Java Code Sample

```
String employee = api.getEntity((short)1, 104);
```

C# Code Sample

```
var employee = api.GetEntity(1, 104);
```

Response

```
<?xml  version="1.0" encoding="utf-16"?>
<employee key="104">
  <created>2014-11-16 14:32:21</created>
  <userkey>1</userkey>
  <user>Administrator, System</user>
  <useremployeekey>1</useremployeekey>
  <masterkey>0</masterkey>
  <employeekey>104</employeekey>
  <employeecode>00000</employeecode>
  <firstnameknown>Joe</firstnameknown>
  <lastname>Bloggs</lastname>
  <cellphone>083-555-5555</cellphone>
  <emailaddress>info@ppo.co.za</emailaddress>
  <active>True</active>
  <e01>False</e01>
  <employeejobtitle>264</employeejobtitle>
  <employeedepartment>278</employeedepartment>
  <includeincapacity>True</includeincapacity>
  <standardhours>0</standardhours>
  <chargeout0>100</chargeout0>
  <chargeout1>100</chargeout1>
  <chargeout2>0</chargeout2>
  <birthdate>1986-08-16</birthdate>
  <firstnamefull>Joe</firstnamefull>
  <initials>J</initials>
  <title>Mr</title>
  <postaladdress>235 Sunnighill, Johannesburg</postaladdress>
  <physicaladdress>5 Sunnighill, Johannesburg</physicaladdress>
  <homephone>(011) 223 5522</homephone>
  <workphone>(011) 336 2258</workphone>
  <extension>2258</extension>
  <e02>True</e02>
  <allfields>
    ...
  </allfields>
</employee>
```

Exceptions

PPO.UnableToRetrieveRecordException

PPO.UnableToRetrieveRecordException: Unable to retrieve record from the database [Object: Employee, EmployeeKey: 0].

Cause

The value specified for the entityKey parameter has no valid entity associated with it.

Solution

Specify a valid entity key for the given entity type. The entity key can be retrieved when doing list retrievals or by checking the URL in PPO when navigated to the entity.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.2 GetEntities

Retrieves XML that contains all the details for all entities belonging to the specified type.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.

Returns

XML that represents entities with all their details.

Example

To retrieve a list of all health indicators.

Java Code Sample

```
String healthIndicators = api.getEntities((short)12);
```

C# Code Sample

```
var healthIndicators = api.GetEntities(12);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<healthindicatorlist totalrowcount="44" startindex="0">
  <healthindicator key="112">
    <created>2014-11-21 14:45:40</created>
    <userkey>1</userkey>
    <user>Administrator, System</user>
    <useremployeekey>1</useremployeekey>
    <masterkey>0</masterkey>
    <healthindicatorkey>112</healthindicatorkey>
    <healthindicatorprojectkey>83</healthindicatorprojectkey>
    <healthindicatortitle>Issues</healthindicatortitle>
    <healthindicatordesc></healthindicatordesc>
    <healthindicatorrag>3</healthindicatorrag>
    <healthindicatoraction>Managed closely.</healthindicatoraction>
    <allfields>
      <field fieldname="HealthIndicatorKey">
        <value>112</value>
        <displayvalue>112</displayvalue>
        <fieldtype>100</fieldtype>
        <hidden>False</hidden>
      </field>
      <field fieldname="HealthIndicatorProjectKey">
        <value>83</value>
        <displayvalue>Sales Recovery Pilot</displayvalue>
        <fieldtype>101</fieldtype>
        <hidden>False</hidden>
      </field>
      <field fieldname="HealthIndicatorTitle">
        <value>Issues</value>
        <displayvalue>Issues</displayvalue>
        <fieldtype>1</fieldtype>
        <hidden>False</hidden>
      </field>
      <field fieldname="HealthIndicatorDesc">
        <value></value>
        <displayvalue></displayvalue>
        <fieldtype>12</fieldtype>
        <hidden>False</hidden>
      </field>
      <field fieldname="HealthIndicatorRAG">
        <value>3</value>
        <displayvalue>3</displayvalue>
        <fieldtype>14</fieldtype>
        <hidden>False</hidden>
      </field>
      <field fieldname="HealthIndicatorAction">
        <value>Managed. No current threat.</value>
        <displayvalue>Managed. No current threat.</displayvalue>
        <fieldtype>12</fieldtype>
        <hidden>False</hidden>
      </field>
    </allfields>
    <documentlinks></documentlinks>
  </healthindicator>
</healthindicatorlist>
```

Exceptions

No unique exceptions for this method.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.3 GetEntitiesByFilterKey

Retrieves XML for all entities of the specified type, filtered by the filter specified by the filter key.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.
filterKey	Short	A unique key that identifies a filter created in PPO. This specific filter will be applied to the list of entities being retrieved. A filter's key can be obtained by clicking on the edit filter icon and then checking the URL of the pop-up page for the key parameter.

Returns

XML that represents entities with all their details.

Example

To retrieve a list of all the employees that are currently active on PPO, use the 'Only Active Employees' filter key.

Java Code Sample

```
String activeEmployees = api.getEntitiesByFilterKey((short)1, (short)15);
```

C# Code Sample

```
var activeEmployees = api.GetEntitiesByFilterKey(1, 15);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<employeelist totalrowcount="16" startindex="0">
  <employee key="107">
    <created>2014-11-21 13:43:05</created>
    <userkey>1</userkey>
    <user>Administrator, System</user>
    <useremployeekey>1</useremployeekey>
    <masterkey>0</masterkey>
    <employeekey>107</employeekey>
    <employeecode>DEMO</employeecode>
    <firstnameknown>Tracey</firstnameknown>
    <lastname>Marais</lastname>
    <cellphone>083-456-2341</cellphone>
    <emailaddress>tracey.marais@regal.com</emailaddress>
    <active>True</active>
    <e01>False</e01>
    <employeejobtitle>270</employeejobtitle>
    <employeedepartment>278</employeedepartment>
    <includeincapacity>True</includeincapacity>
    <standardhours>0</standardhours>
    <chargeout0>360</chargeout0>
    <chargeout1>300</chargeout1>
    <chargeout2>0</chargeout2>
    <birthdate>1978-08-25</birthdate>
    <firstnamefull>Tracey Jane</firstnamefull>
    <initials>T J</initials>
    <title>Miss</title>
    <postaladdress>P O Box 4632, Southdale, 4018</postaladdress>
    <physicaladdress>10 Riverpark Avenue, Southdale, 4018</physicaladdress>
    <homephone>(011) 678-2061</homephone>
    <workphone>(011) 666 8888</workphone>
    <extension>8888</extension>
    <e02>False</e02>
    <allfields>
      <field fieldname="EmployeeKey">
        <value>107</value>
        <displayvalue>Marais, Tracey</displayvalue>
        <fieldtype>100</fieldtype>
        <hidden>False</hidden>
      </field>
      ...
    </employeelist>
</healthindicatorlist>
```

Exceptions

PPO.ObjectDoesNotExistException

PPO.ObjectDoesNotExistException: Filter does not exist [FilterKey: -1].

Cause

The error occurs when any value specified for the filter key is not associated with an existing filter in PPO.

Solution

Specify a valid key for a filter that exists. A filter's key can be obtained by clicking on the edit filter icon and then checking the URL of the pop-up page for the key parameter.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.4 GetEntitiesFiltered

Retrieves XML for all entities of the specified type where the field name matches the specified value.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.
fieldName	String	Name of the field you want to filter on. To find a field name: In PPO, go to Administration and then Data Fields Select the appropriate entity from the drop down The name will appear in the 'Field Name' column
fieldValue	String	The value that the specified field name must be equal to for an entity to be included in the list that will be returned.

Returns

XML that represents entities with all their details.

Example

To retrieve a list of all employees with the first name of Harry.

Java Code Sample

```
String employee = api.getEntitiesFiltered((short)1, "FirstNameKnown", "Harry");
```

C# Code Sample

```
  var employee = api.GetEntitiesFiltered(1, "FirstNameKnown", "Harry");
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<employeelist totalrowcount="16" startindex="0">
  <employee key="107">
    <created>2014-11-21 13:43:05</created>
    <userkey>1</userkey>
    <user>Administrator, System</user>
    <useremployeekey>1</useremployeekey>
    <masterkey>0</masterkey>
    <employeekey>107</employeekey>
    <employeecode>DEMO</employeecode>
    <firstnameknown>Tracey</firstnameknown>
    <lastname>Marais</lastname>
    <cellphone>083-456-2341</cellphone>
    <emailaddress>tracey.marais@regal.com</emailaddress>
    <active>True</active>
    <e01>False</e01>
    <employeejobtitle>270</employeejobtitle>
    <employeedepartment>278</employeedepartment>
    <includeincapacity>True</includeincapacity>
    <standardhours>0</standardhours>
    <chargeout0>360</chargeout0>
    <chargeout1>300</chargeout1>
    <chargeout2>0</chargeout2>
    <birthdate>1978-08-25</birthdate>
    <firstnamefull>Tracey Jane</firstnamefull>
    <initials>T J</initials>
    <title>Miss</title>
    <postaladdress>P O Box 4632, Southdale, 4018</postaladdress>
    <physicaladdress>10 Riverpark Avenue, Southdale, 4018</physicaladdress>
    <homephone>(011) 678-2061</homephone>
    <workphone>(011) 666 8888</workphone>
    <extension>8888</extension>
    <e02>False</e02>
    <allfields>
```

```
      <field fieldname="EmployeeKey">
        <value>107</value>
        <displayvalue>Marais, Tracey</displayvalue>
        <fieldtype>100</fieldtype>
        <hidden>False</hidden>
      </field>
      ...
    </employeelist>
```

Exceptions

PPO.UnableToRetrieveFieldDetailsException

PPO.UnableToRetrieveFieldDetailsException: Unable to retrieve field details [Field name: sdgvgerrvgervg].

Cause

Any value specified for the fieldName parameter that is not an exact match of one of the fields of the given entity type will result in this error.

Solution

Make sure what the field names are for the entity type being retrieved and specify a valid one for filtering.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.5 GetEntitiesCompressed

Retrieves XML for all entities of the specified type (the same as the GetEntities web method), but returns a ZIP file as a byte array which contains the XML. This method will always be faster than the GetEntities method, because the size of the response will be about 15% of the size of the non-compressed method.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.

Returns

A ZIP file of XML that represents entities with all their details.

Example

To retrieve a large number of tasks, consider using this web method.

Java Code Sample

```
byte[] compressedTasks = api.getEntitiesCompressed((short)5);
```

C# Code Sample

```
byte[] compressedTasks = api.GetEntitiesCompressed(5);
```

Example Response

A ZIP file in the form of an array of bytes. The file contains a list of all tasks in XML format as described in the GetEntities web method.

Exceptions

Please refer to the GetEntities web method as the exceptions are the same as for the GetEntitiesCompressed web method.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.6 GetEntitiesByFilterKeyCompressed

Retrieves XML for all entities of the specified type filtered by the filter key specified (the same as the GetEntitiesByFilterKey web method), but returns a ZIP file as a byte array which contains the XML. This method will always be faster than the GetEntitiesByFilterKey method, because the size of the response will be about 15% of the size of the non-compressed method.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.
filterKey	Short	A unique key that identifies a filter created on PPO. This specific filter will be applied to the list of entities being retrieved. A filter's key can be obtained by clicking on the edit filter icon and then checking the URL of the pop-up page for the key parameter.

Returns

A ZIP file of XML that represents entities with all their details.

Example

To retrieve a large number of employees, but applying the 'Only Active Employees' filter (filter key 15).

Java Code Sample

```
byte[] activeEmployees = api.getEntitiesByFilterKeyCompressed((short)1, (short)15);
```

C# Code Sample

```
byte[] activeEmployees = api.GetEntitiesByFilterKeyCompressed(1, 15);
```

Example Response

A ZIP file in the form of an array of bytes. The file contains a list of all employees currently active on the instance in XML format.

Exceptions

Please refer to the GetEntitiesByFilterKey web method as the exceptions are the same as for the GetEntitiesByFilterKeyCompressed web method.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.7 GetEntitiesFilteredCompressed

Retrieves XML for all entities of the specified type where the field name matches the specified value (the same as the GetEntitiesFiltered web method), but returns a ZIP file as a byte array which contains the XML. This method will always be faster than the GetEntitiesFiltered, because the size of the response will be about 15% of the size of the non-compressed method.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.
fieldName	String	Name of the field you want to filter upon. To find a field name: In PPO, go to Administration and then Data Fields Select the appropriate entity from the drop down The name will appear in the 'Field Name' column
fieldValue	String	The value that the specified field name must be equal to for an entity to be included in the list that will be returned.

Returns

A ZIP file of XML that represents entities with all their details.

Example

To retrieve a list of all tasks with type 'Summary Task'.

Java Code Sample

```
byte[] tasks = api.getEntitiesFilteredCompressed((short)5, "TaskType", "99");
```

C# Code Sample

```
byte[] tasks = api.GetEntitiesFilteredCompressed(5, "TaskType", "99");
```

Example Response

A ZIP file in the form of an array of bytes. The file contains all the details of the project in XML format.

Exceptions

Please refer to the GetEntitiesFiltered web method as the exceptions are the same as for the GetEntitiesFilteredCompressed web method.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.8 GetDeletedEntities

Retrieves XML that contains all the details for deleted entities belonging to the specified type and were delete within the specified date range.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.
startDate	Timestamp	Start date, in UTC, of the date rage within which entities were deleted.
endDate	Timestamp	End date, in UTC, of the date rage within which entities were deleted.

Returns

XML that represents deleted entities with all their details.

Example

To retrieve a list of health indicators deleted in the last two hours.

Java Code Sample

```
Instant instant = Instant.now();

String deletedHealthIndicators = api.getDeletedEntities((short)12, instant.minus(Duration.ofHours(2)),
instant);
```

C# Code Sample

```
var currentDateTime = DateTime.UtcNow;

var deletedHealthIndicators = api.GetDeletedEntities(12, currentDateTime.AddHours(-2),
currentDateTime);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<healthindicatorlist totalrowcount="44" startindex="0">
  <healthindicator key="112">
    <created>2014-11-21 14:45:40</created>
    <userkey>1</userkey>
    <user>Administrator, System</user>
    <useremployeekey>1</useremployeekey>
    <masterkey>112</masterkey>
    <healthindicatorkey>112</healthindicatorkey>
    <healthindicatorprojectkey>83</healthindicatorprojectkey>
    <healthindicatortitle>Issues</healthindicatortitle>
    <healthindicatordesc></healthindicatordesc>
    <healthindicatorrag>3</healthindicatorrag>
    <healthindicatoraction>Managed closely.</healthindicatoraction>
    <allfields>
      <field fieldname="HealthIndicatorKey">
        <value>112</value>
        <displayvalue>112</displayvalue>
        <fieldtype>100</fieldtype>
        <hidden>False</hidden>
      </field>
      ...
    </allfields>
    <documentlinks></documentlinks>
  </healthindicator>
</healthindicatorlist>
```

Exceptions

No unique exceptions for this method.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.9 GetDeletedEntitiesByFilterKey

Retrieves XML for deleted entities of the specified type, filtered by the filter specified by the filter key.

Logical access control

The authenticated user must have 'View' access for the entity type specified e.g. 'View Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that represents an entity type in PPO.
filterKey	Short	A unique key that identifies a filter created in PPO. This specific filter will be applied to the list of entities being retrieved. A filter's key can be obtained by clicking on the edit filter icon and then checking the URL of the pop-up page for the key parameter.

Returns

XML that represents deleted entities with all their details.

Example

To retrieve a list of the deleted costs that were deleted in the last 30 days, first create a "Deleted in the last 30 days" filter and use it as per below example. Assume that the filter key is 41.

Java Code Sample

```
String deletedCosts = api.getDeletedEntitiesByFilterKey((short)9, (short)41);
```

C# Code Sample

```
var deletedCosts = api.GetDeletedEntitiesByFilterKey(9, 41);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<costlist totalrowcount="1" startindex="0">
  <cost key="14">
    <created>2014-11-21 14:45:40</created>
    <userkey>1</userkey>
    <user>System Administrator</user>
    <useremployeekey>1</useremployeekey>
    <masterkey>14</masterkey>
    <displayvalue>OPEX - 2014</displayvalue>
    <costkey>14</costkey>
    <costprojectkey>31</costprojectkey>
    <costcategory>295</costcategory>
    <c01>318</c01>
    <costsubcategory>296</costsubcategory>
    <c02>313</c02>
    <costdescription></costdescription>
    <costbudget>1440</costbudget>
    <costbilled>0</costbilled>
    <costestimate>1440</costestimate>
    <costatcompletion>1440</costatcompletion>
    <costvariance>0</costvariance>
    <costrecoverable>False</costrecoverable>
    <costrecovered>0</costrecovered>
    <allfields>
      <field fieldname="CostKey">
        <value>14</value>
        <displayvalue>14</displayvalue>
        <fieldtype>100</fieldtype>
        <hidden>False</hidden>
        <percentagecolorsetting>0</percentagecolorsetting>
      </field>
      ...
    </allfields>
  </cost>
</costlist>
```

Exceptions

PPO.ObjectDoesNotExistException

PPO.ObjectDoesNotExistException: Filter does not exist [FilterKey: -1].

Cause

The error occurs when any value specified for the filter key is not associated with an existing filter in PPO.

Solution

Specify a valid key for a filter that exists. A filter's key can be obtained by clicking on the edit filter icon and then checking the URL of the pop-up page for the key parameter.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.10 UpdateEntity

Updates or creates an entity based on the XML provided as a parameter. If no key is specified on the XML tag then an entity will be created, otherwise it will update the entity with the specified key. The GetXsd web method can be used to get the schema of an entity. Only the fields that need to be updated should be included in the XML.

Logical access control

The authenticated user must have 'Add' and 'Edit' access for the entity type provided in the entity XML e.g. 'Add Project'. User must also have access to specific item based on your user group filter.

Parameter

Name	Data type	Description
entityXml	String	A string in XML format that represents the entity and its details.

Returns

XML that represents an entity with all its details.

Example

To change the first and last name of an existing employee.

XML Sample

```
<employee key="108">
  <firstnameknown>Joe</firstnameknown>
  <lastname>Bloggs</lastname>
</employee>
```

Java Code Sample

```
//Create XML string that represents an entity

api.updateEntity(employeeXmlString);
```

C# Code Sample

```
XElement employee = new XElement("employee",
                new XAttribute("key", 108),
                new XElement("firstnameknown", "Joe"),
                new XElement("lastname", "Bloggs"));

api.UpdateEntity(employee.ToString());
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<employee key="108">
  <created>2015-01-16 10:22:52</created>
  <userkey>4</userkey>
  <user>Adams, Harry</user>
  <useremployeekey>112</useremployeekey>
  <masterkey>0</masterkey>
  <employeekey>108</employeekey>
  <employeecode>CLOSE</employeecode>
  <firstnameknown>Joe</firstnameknown>
  <lastname>Bloggs</lastname>
  <cellphone>082-465-0283</cellphone>
  <emailaddress>harry.waczkow@regal.com</emailaddress>
  <active>True</active>
  <e01>False</e01>
  <employeejobtitle>255</employeejobtitle>
  <employeedepartment>280</employeedepartment>
  <includeincapacity>True</includeincapacity>
  <standardhours>0</standardhours>
  <chargeout0>520</chargeout0>
  <chargeout1>470</chargeout1>
  <chargeout2>0</chargeout2>
  <birthdate>1974-12-22</birthdate>
  <firstnamefull>Harold Anthony</firstnamefull>
  <initials>H A</initials>
  <title>Mr</title>
  <postaladdress>P O Box 35501, Windsor West, 2115</postaladdress>
  <physicaladdress>8 Countesses Manor, Countesses Street, Windsor West, 2115</physicaladdress>
  <homephone>(012) 678-6472</homephone>
  <workphone>(012) 362-5010</workphone>
  <extension>5010</extension>
  <e02>False</e02>
  <allfields>
  ...
</employee>
```

Exceptions

PPO.InvalidXmlException

PPO.InvalidXmlException: The XML is not valid.

Cause

The XML passed as a parameter is an empty string or the XML is invalid.

Solution

Make sure the XML being passed uses the XML schema required for the given entity. You can get the XML schema by using the GetXsd web method.

PPO.UnableToRetrieveRecordException

PPO.UnableToRetrieveRecordException: Unable to retrieve record from the database [Object: Employee, EmployeeKey: 34].

Cause

This exception only applies to updates. The key specified in the XML does not match any entity of that type with the same key.

Solution

Specify a valid entity key for the given entity type. The entity key can be retrieved when doing list retrievals or by checking the URL in PPO when navigated to the entity.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.11 UpdateEntities

Updates or creates entities based on the XML provided as a parameter. This method is the same as the UpdateEntity, but instead it allows batch inserts or updates. Use the GetXsd method to get the schemas for different entities. The outer tag of the XML is not important. This method can also be used to update different entity types in the same batch.

Logical access control

The authenticated user must have 'Add' and 'Edit' access for the entity type provided in the

entity XML e.g. 'Add Project'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityXml	String	A string in XML format that represents the entities and their details. The XML will be used to create new entities when keys are not provided and will update existing entities when keys are present.

Returns

A number that indicates how many entities were updated or created.

Example

To update multiple tasks with different values.

Request

```
<tasks>
  <task key="412">
    <tasktype>30</tasktype>
    <actualprogress>50</actualprogress>
  </task>
  <task key="415">
    <taskname>Edited task</taskname>
    <actualprogress>65</actualprogress>
  </task>
</tasks>
```

Java Code Sample

```
//Create XML string that represents an entity

api.updateEntities(tasksXmlString);
```

C# Code Sample

```
XElement tasks = new XElement("tasks",
                 new XElement("task",
                     new XAttribute("key", 412),
                     new XElement("tasktype", 30),
                     new XElement("actualprogress", 50)),
                 new XElement("task",
                     new XAttribute("key", 415),
                     new XElement("taskname", "Edited task"),
                     new XElement("actualprogress", 65)));
var tasksXml = api.UpdateEntities(tasks.ToString());
```

Example Response

UpdateEntities responds with the number of entities updated or created, in this case it will respond with the number 2. If there is an exception on one of the entities you are trying to update or create, the response will be the exception, but the entities that were specified before the entity that caused the exception will still be created/updated.

Exceptions

PPO.InvalidXmlException

PPO.InvalidXmlException: The XML is not valid.

Cause

The XML passed as a parameter is an empty string or the XML is not valid.

Solution

Make sure the XML being passed uses the XML schema required for the given entity. You can get the XML schema by using the GetXsd web method.

PPO.UnableToRetrieveRecordException

PPO.UnableToRetrieveRecordException: Unable to retrieve record from the database [Object: Employee, EmployeeKey: 34].

Cause

This exception only applies to updates. A key specified in the XML does not match any entity of that type with the same key. The entity type and the key of the one that caused the exception will be specified in the error message.

Solution

Specify a valid entity key for the given entity type. The entity key can be retrieved when doing list retrievals or by checking the URL in PPO when navigating to the entity.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here…

4.12 UpdateDocumentEntity

Updates or creates a document entity based on the XML and file provided. Similar to UpdateEntity, but accompanied by a file.

Logical access control

The authenticated user must have 'Add' and 'Edit' access for the document entity type e.g. 'Add Document'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityXml	String	A string in XML format that represents the entity and its details. The XML will be used to create a new entity when a key is not provided and will update an existing entity when a key is present.
fileData	Byte array	The physical file you want to upload to the instance as an update or as a new document.

Returns

XML that represents the document entity with all its details.

Example

To add a sales proposal document to project 66. Not only will the document be uploaded but it will be associated with a document entity which can be attached to a project etc.

XML Sample

```
<document>
  <documentprojectkey>66</documentprojectkey>
  <documenttitle>Sales Proposal</documenttitle>
  <documenttype>241</documenttype>
  <documentfilename>Sales Proposal.docx</documentfilename>
</document>
```

Java Code Sample

```
//Create XML string that represents an entity

String filePath = "C:\\Users\\JoeBloggs\\Desktop\\SalesProposal.docx";
byte[] file = Files.readAllBytes(Paths.get(filePath));
String documentXmlResponse = api.updateDocumentEntity(documentXml, file);
```

C# Code Sample

```
XElement document = new XElement("document",
                    new XElement("documentprojectkey", 66),
                    new XElement("documenttitle", "Sales Proposal"),
                    new XElement("documenttype", 241),
                    new XElement("documentfilename", "Sales Proposal.docx"));

byte[] file = File.ReadAllBytes(@"C:\Users\JoeBloggs\Desktop\SalesProposal.docx”));
var documentXml = api.UpdateDocumentEntity(document.ToString(), file);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<document key="109">
  <created>2015-01-16 13:47:19</created>
  <userkey>4</userkey>
  <user>Adams, Harry</user>
  <useremployeekey>112</useremployeekey>
  <masterkey>0</masterkey>
  <documentkey>109</documentkey>
  <documentprojectkey>66</documentprojectkey>
  <documenttitle>Sales Proposal</documenttitle>
  <documentdescription></documentdescription>
  <documentfilename>4af04e22-f23d-41b8-95ba-ef763c5f4b48</documentfilename>
  <documenttype>241</documenttype>
  <documentstatus></documentstatus>
  <documentcheckedout>False</documentcheckedout>
  <allfields>
    ...
  </allfields>
</document>
```

Exceptions

PPO.InvalidXmlException

PPO.InvalidXmlException: The XML is not valid.

Cause

The XML passed as a parameter is an empty string or the XML is not valid.

Solution

Make sure the XML being passed uses the XML schema required for the given entity. You can get the XML schema by using the GetXsd web method.

PPO.UnableToRetrieveRecordException

PPO.UnableToRetrieveRecordException: Unable to retrieve record from the database [Object: Document, DocumentKey: 34].

Cause

This exception only applies to updates. A key specified in the XML does not match any document with the same key.

Solution

Specify a valid document key or leave the key out of the XML to upload a new document.

System.Web.HttpException

System.Web.HttpException: Maximum request length exceeded.

Cause

This error occurs when trying to upload a file larger than the limit allowed. Refer to this knowledge base article for more information.

Solution

Only upload files smaller in size than the limit specified in the FAQ referenced above. ZIP files are allowed so it is best to compress large files.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.13 LinkDocumentEntity

Link a document to a related entity.

Logical access control

The authenticated user must have 'Edit' access for the entity type provided in the entity XML e.g. 'Edit Task'. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
entityType	Short	A numeric value that specifies the type of PPO entity you want to link to.
entityKey	Integer	The unique key of the specific entity. (EmployeeKey, TaskKey, etc.)
documentKey	Integer	The unique key that identifies the document you want to link to the entity.

Returns

Nothing.

Example

To link a sales proposal to the task it relates to. The document and task should already exist if you want to link it.

Java Code Sample

```
api.linkDocumentEntity((short)5, 70, 110);
```

C# Code Sample

```
api.LinkDocumentEntity(5, 70, 110);
```

Example Response

No response.

Exceptions

PPO.UnableToRetrieveRecordException

PPO.UnableToRetrieveRecordException: Unable to retrieve record from the database [Object: Employee, EmployeeKey: 0].

Cause

The value specified for the entityKey parameter is not valid and no such key is associated with an entity.

OR

The entity type exists and the entity key is associated with an existing entity, but the document that you are trying to link does not exist.

Solution

Specify a valid entity key for the given entity type. The entity key can be retrieved when doing list retrievals or by checking the URL in PPO when navigated to the entity.

OR

Specify a valid document key of a document that is already uploaded to PPO.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.14 GetCustomListItemKey

Retrieves a custom list item key by description.

Logical access control

No access control for this method.

Parameters

Name	Data type	Description
customListKey	Integer	Unique key identifying the custom list from which an item matching the description must be returned. You can get custom list key by navigating to the Administration tab in PPO then clicking the Custom Lists icon and then selecting a custom list from the list. The custom list's key is in the URL under the key parameter of the query string.
description	String	The description of the custom list item. You can get an item's description by selecting the custom list in PPO and then the custom list items are displayed with their respective descriptions.

Returns

The key of the custom list item.

Example

Get the key of the 'Milestone' custom list item.

Java Code Sample

```
int itemKey = api.getCustomListItemKey((short)8, "Milestone");
```

C# Code Sample

```
var itemKey = api.GetCustomListItemKey(8, "Milestone");
```

Example Response

30

Exceptions

PPO.ObjectDoesNotExistException

PPO.ObjectDoesNotExistException: Milestone does not exist [strListItemDescription: -100].

Cause

This error can be caused by specifying a custom list that does not exist or by specifying a custom list item description that does not match any item related to the given custom list key.

Solution

Firstly make sure the custom list key is associated with an existing custom list and then check the item's description.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.15 GetCustomListItem

Retrieves a custom list item for the specified key.

Logical access control

No access control for this method.

Parameters

Name	Data type	Description
itemKey	Integer	The unique key of the custom list item that you want to retrieve. You can get an item's key by selecting the custom list in PPO and then selecting the custom list item and then getting the key from the URL.

Returns

XML that represents the custom list item with all its details.

Example

When you have a custom list item's key and you want to know what it represents.

Java Code Sample

```
String customListItemXml = api.getCustomListItem((short)30);
```

C# Code Sample

```
var customListItem = api.GetCustomListItem(30);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<customlistitem key="30" code="MST">
  <customlistkey>8</customlistkey>
  <description>Milestone</description>
  <sortkey>4</sortkey>
  <protected>True</protected>
</customlistitem>
```

Exceptions

PPO.ObjectDoesNotExistException

PPO.ObjectDoesNotExistException: Custom List Item does not exist [intListItemKey: -100].

Cause

Any value specified for the key of the custom list item that is not associated with an existing item will result in this error.

Solution

Make sure the custom list item key supplied is valid.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.16 GetCustomListItems

Retrieves custom list items for the specified key.

Logical access control

No access control for this method.

Parameters

Name	Data type	Description
customListKey	Integer	Unique key identifying the custom list from which an item matching the description must be returned. You can get custom list key by navigating to the Administration tab in PPO then clicking the Custom Lists icon and then selecting a custom list from the list. The custom list's key is in the URL under the key parameter of the query string.

Returns

XML that represents the custom list items with all their details.

Example

When you want to retrieve a list of all project statuses.

Java Code Sample

```
String customListItemsXml = api.getCustomListItems((short)7);
```

C# Code Sample

```
var items = api.GetCustomListItems(7);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<customlistitems>
  <customlistitem key="26" code="">
    <customlistkey>7</customlistkey>
    <description>Proposed</description>
    <sortkey>1</sortkey>
    <protected>False</protected>
  </customlistitem>
  <customlistitem key="67" code="">
    <customlistkey>7</customlistkey>
    <description>Active</description>
    <sortkey>2</sortkey>
    <protected>False</protected>
  </customlistitem>
  <customlistitem key="68" code="">
    <customlistkey>7</customlistkey>
    <description>On Hold</description>
    <sortkey>3</sortkey>
    <protected>False</protected>
  </customlistitem>
  <customlistitem key="69" code="">
    <customlistkey>7</customlistkey>
    <description>Rejected</description>
    <sortkey>4</sortkey>
    <protected>False</protected>
  </customlistitem>
  <customlistitem key="78" code="">
    <customlistkey>7</customlistkey>
    <description>Closed</description>
    <sortkey>5</sortkey>
    <protected>False</protected>
  </customlistitem>
</customlistitems>
```

Exceptions

PPO.ObjectDoesNotExistException

PPO.ObjectDoesNotExistException: Custom List does not exist [CustomListKey: 1000].

Cause

Any value specified for the key of the custom list that is not associated with an existing custom list will result in this error.

Solution

Make sure the custom list key is associated with an existing custom list.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.17 GetCustomList

Retrieves a custom list for the specified key.

Logical access control

No access control for this method.

Parameters

Name	Data type	Description
customListKey	Integer	Unique key identifying the custom list from which an item matching the description must be returned. You can get custom list key by navigating to the Administration tab in PPO then clicking the Custom Lists icon and then selecting a custom list from the list. The custom list's key is in the URL under the key parameter of the query string.

Returns

XML that represents the custom list with all its details.

Example

To retrieve the description for a particular custom list.

Java Code Sample

```
String customListXml = api.getCustomList((short)13);
```

C# Code Sample

```
var customList = api.GetCustomList(13);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<customlist key="13" code="">
  <name>Risk - Impact</name>
  <description>Used on risk entity to determine the impact.</description>
  <autosort>False</autosort>
  <protected>True</protected>
</customlist>
```

Exceptions

PPO.ObjectDoesNotExistException

PPO.ObjectDoesNotExistException: Custom List does not exist [CustomListKey: 1000].

Cause

Any value specified for the key of the custom list that is not associated with an existing custom list will result in this error.

Solution

Make sure the custom list key is associated with an existing custom list.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.18 GetCustomLists

Retrieves all custom lists.

Logical access control

No access control for this method.

Parameters

No parameters.

Returns

XML that represents the custom lists with all their details.

Example

To retrieve a list of all custom lists.

Java Code Sample

```
String customListsXml = api.getCustomLists();
```

C# Code Sample

```
var customLists = api.GetCustomLists();
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<customlists>
  <customlist key="101" code="AS">
    <name>Approval Status</name>
    <description>Approval Status</description>
    <autosort>False</autosort>
    <protected>True</protected>
  </customlist>
  <customlist key="20" code="">
    <name>Comment - Type</name>
    <description>Used on the comments entity to determine the type of comment.</description>
    <autosort>False</autosort>
    <protected>True</protected>
  </customlist>
  ...
</customlists>
```

Exceptions

No unique exceptions for this web method.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.19 GetReportBytes

PPO provides functionality for creating reports. Refer to this knowledge base article for more information. This method returns a report on PPO as a stream of bytes (download of report). The report can be downloaded as a PDF by adding a query parameter of display with a value of pdf or a value of excel for a spreadsheet. The report can also be in a datasheet view by adding a query string parameter of mode and a value of data.

Logical access control

The user must have 'View' access for the report. User must also have access to specific item based on your user group filter.

Parameters

Name	Data type	Description
url	String	The URL of the report that you want to retrieve. To get this URL, run the report and copy the URL from the address bar.

Returns

The requested report as an array of bytes.

Example

A copy of a report needs to be saved locally and the GetReportBytes web method allows you to download a report as an array of bytes that can be stored.

Java Code Sample

```
byte[] report =
api.getReportBytes("https://www.ppolive.com/acme/configurablereports/reportSectionalEntity.aspx?ProjectList=EntityKey~74&reportingItemType=Report&reportingItemKey=502");
```

C# Code Sample

```
byte[] report =
api.GetReportBytes("https://www.ppolive.com/acme/configurablereports/reportSectionalEntity.aspx?ProjectList=EntityKey~74&reportingItemType=Report&reportingItemKey=502");
```

Example Response

An array of bytes that is the physical report in html format. You can receive the report in pdf format by specifying the following query string parameter: display=pdf.

Exceptions

PPO.InvalidUrlException

PPO.InvalidUrlException: The Url is not in a valid format.

Cause

The URL given as a parameter is not a valid URL and does not comply with any URL conventions.

Solution

The given URL cannot be correct. Go on PPO to the Reports tab, then select the appropriate report and view it in report view, then copy the URL directly. ***PPO.InvalidReportUrlException***

PPO.InvalidReportUrlException: The specified URL does not point to a valid PPO resource.

Cause

When a valid URL is provided, but it doesn't point to a report on your own PPO instance.

Solution

Go on PPO to the Reports tab, then select the appropriate report and view it in report view, then copy the URL directly.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

4.20 GetXsd

All submitted XML needs to comply with a specific XML schema that can be generated by calling the GetXsd web method. The schema of entities are different for each instance, because not every instance uses the same data fields for the entities and custom entities are also uniquely configured for a specific instance.

Logical access control

No access control for this method.

Parameters

Name	Data type	Description
entityType	Integer	A numeric value that represents an entity type in PPO.

Returns

XML that indicates the schema that must be used with the entity you specified.

Example

Prior to an updating a risk, you want to validate the XML data against the required schema for the risk entity.

Java Code Sample

```
String riskEntityXsd = api.getXsd(7);
```

C# Code Sample

```
var riskEntityXsd = api.GetXsd(7);
```

Example Response

```
<?xml version="1.0" encoding="utf-16"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="risklist">
    <xsd:complexType>
      <xsd:sequence minOccurs="0">
        <xsd:element name="risk" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="r01" type="xsd:date" nillable="true" />
              <xsd:element name="r02" type="xsd:decimal" />
              <xsd:element name="r03" type="xsd:integer" />
              <xsd:element name="r04" type="xsd:decimal" />
              <xsd:element name="r07" type="xsd:date" nillable="true" />
              <xsd:element name="riskaction" type="xsd:string" />
              <xsd:element name="riskassignedto" type="xsd:integer" />
              <xsd:element name="riskcategory" type="xsd:integer" />
              <xsd:element name="riskcontents" type="xsd:string" />
              <xsd:element name="riskimpact" type="xsd:integer" />
              <xsd:element name="riskkey" type="xsd:integer" />
              <xsd:element name="riskprobability" type="xsd:integer" />
              <xsd:element name="riskprojectkey" type="xsd:integer" />
              <xsd:element name="riskrag">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:integer">
                    <xsd:pattern value="0|1|2|3" />
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
              <xsd:element name="riskrating" type="xsd:decimal" />
              <xsd:element name="riskstatus" type="xsd:integer" />
              <xsd:element name="risktitle">
                <xsd:simpleType>
                  <xsd:restriction base="xsd:string">
                    <xsd:maxLength value="200" />
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:element>
            </xsd:sequence>
            <xsd:attribute name="key" type="xsd:integer" />
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Exceptions

No unique exceptions for this method.

Refer to the 'General Exceptions' section at the end of the document for any exceptions not listed here.

5 General exceptions

System.ArgumentNullException

System.ArgumentNullException: Value cannot be null.

Cause

This error occurs when one or more required parameters for a web method are not specified and are left null.

Solution

Specify values for all required parameters when invoking any web method.

PPO. EntityTypeNotInUseException

PPO. EntityTypeNotInUseException: The value specified for entityType is not in use [EntityType: 100].

Cause

This error occurs when any entityType value is specified which is not in use currently.

Solution

A numeric value that represents an entity type in PPO.
To find the key for an entity type:

- In PPO, go to Administration and then Data Fields.
- Select the appropriate entity from the drop down.
- The key will appear in the address bar.

PPO.NoAccessException

PPO.NoAccessException: Members of the 'Project Managers' user group do not have access to this resource [Key: 1 Description: foo].

Cause

Will occur when the user that is authenticated is part of a user group that does not have access to view an entity type or when the user does not have access to a specific item.

Solution

Either get access to view the entity from an administrator authenticate with another user account.

PPO.CustomValidationException

PPO.CustomValidationException: {custom message of the validation rule e.g. 'You are not allowed to change an employee's birth date'}.

Cause

You will receive this exception when a validation rule that is configured on PPO fails.

Solution

Either change the details of your data being submitted to conform to the validation rules, change the validation rules if it is practical or add an exception on the business rule. Refer to this knowledge base article for more information.

PPO.AuthenticationFailureException

PPO.AuthenticationFailureException: User not authenticated.

Cause

This error occurs when a web method is called without being authenticated first.

Solution

Refer to the **Authentication and LogInExplicit** section and also refer to the **Session state and cookie container** section for help.

6 Not Supported

The following web methods are no longer supported but exist for backward compatibility reasons:

FlushCachedItem

GetCacheSize

GetCacheSize

GetCacheTypeSize

GetCacheCountAll

GetCacheCount

GetCacheTypeCount

GetActiveUserCount

ForceGarbageCollect

UploadFile

FlushCache

UpdateDocumentEntityWithUserKey

SendApprovalEmail

CreateItemsFromTemplateProject