PEARSON ALWAYS LEARNING

# EQUELLA®

# PHP SOAP Tutorial

*Version 6.4*

Document History

| Document No. | Reviewed | | Finalised | | Published | |
|---|---|---|---|---|---|---|
| 1 | 22/05/2015 | | 22/05/2015 | | 22/05/2015 | |

May 2015 edition.

Property of:

Pearson
Level 1, 11 Elizabeth Street
Hobart, Australia, 7000

Contact documentation@equella.com for matters relating to documentation.

## Table of Contents

# PHP SOAP tutorial

## Running the examples

You need to have the SOAP extension enabled in your PHP installation to use these samples. To do this, make sure you have extension=php_soap.dll (Windows) or extension=php_soap.so (UNIX) in your **php.ini** file. Restart your web server after making this change. You may need to download the appropriate extension file if you do not have it. Also make sure that your PHP installation has ‘magic quotes’ disabled ( http://www.php.net/manual/en/security.magicquotes.disabling.php )

To run the example files, modify settings.php with the appropriate values (SOAP endpoint, username and password). The SOAP endpoint is of the format: *http://INSTITUTION_URL/services/SoapService50*

For example: *http://mydomain.edu.au/myinst/services/SoapService50*

Feel free to customise or extend the sample files for your own purposes.

## Guide to the example code

The example code contains some helper classes (EQUELLA and XMLWrapper) which you are free to use and modify. These classes have been created to simplify the processing of the XML results and XML parameters returned from and supplied to the SOAP methods. The EQUELLA and XMLWrapper classes are not required to use the EQUELLA SOAP service, they are simply for convenience.

### Search

This example shows how you can retrieve a list of items from EQUELLA based on a few search parameters. Most of the code in the search.php file is dedicated to displaying the form; the actual SOAP invocation code is clearly marked ‘*EQUELLA SOAP Searching Code*’ at the bottom of the file. Here is a summary of that code:

- It creates an EQUELLA object which is a thin wrapper around the PHP SoapClient object.
- It invokes searchItems on the EQUELLA object. This is also a thin wrapper around the `SoapClient->searchItems()` method. This method returns an XMLWrapper object for convenience, rather than raw XML string returned by the SoapClient.
- It retrieves information from the XML, such as the number of results returned, the number of results available and the actual result list. The format of the returned XML and the meanings/values of the parameters are documented in the *SOAP API* documentation.

### Contribute

This example shows how you can contribute an item to a collection and give it a name, description and a file attachment. As with the search example, much of the contribute.php code is dedicated to displaying the form. The contribution SOAP code is clearly marked '*EQUELLA SOAP Contribution Code'* towards the bottom of the file. Here is a summary of that code:

- It invokes newItem on the previously created EQUELLA object (the EQUELLA object was actually used earlier to display the list of collections). Invoking newItem will create a new 'staging' area on the server for this item while it is being edited. The item will *not* be created on the server until saveItem is called, however you should edit the details of the item before doing so, as this example does.
- It sets the name and description of the item by setting the values of the XML
- It determines if a file has been uploaded and if so, it uploads the file to the staging area. Note that this does not create an attachment on the item, it simply uploads the file. The uploadFile method on the EQUELLA class converts the binary file into a base 64 encoded string. You would need to do this yourself if using the raw SoapClient method.
- It creates an attachment node for the uploaded file on the item XML. The format of the attachment node is documented in the *Item XML format* section of the *SOAP API* documentation.
- It invokes the saveItem method with a submit value of '*1*'. A submit value of '*1*' means that the item will now either go live or move onto the next moderation step, a value of '*0*' means the item will be saved with a status of '*draft*'.

# Writing your own code

In general the process for invoking a SOAP method is as follows:

1. Create a SoapClient object.

When creating a SoapClient object, you MUST also specify the optional 'location' parameter. The WSDL files generated by the EQUELLA server have an incorrect URL within them.

2. Call the login method.

The parameter names when invoking SOAP methods are '*in0*', '*in1*', '*in2*' etc and values for all of these need to be specified. *(Note that all Boolean parameters should be specified as '0' or '1'*) For example, to invoke the login method:

```
$client->login(array('in0' => $username, 'in1' => $password));
```

The user that you login as must have sufficient privileges to perform your required task. You cannot do anything over SOAP that you cannot do within the EQUELLA Digital Repository .

3. Invoke a method:

The results of all SOAP methods (if available) will always be in a field called '*out'*.

E.g.:

```
$collections = $client->getContributableCollections()->out;
```

*(NOTE: All SOAP methods will return either a simple string, number or an XML object. We recommend using the PHP DOMDocument and DOMXPath classes to parse the results of any SOAP call).*

4. Call the logout method

This is not strictly necessary, the user session will eventually time out on the server, but it is good practice to call logout to free server side resources.

# Contact Client Support

We are always happy to help.

If your organisation has a support agreement with EQUELLA then help is available at http://equella.custhelp.com.