PEARSON ALWAYS LEARNING

EQUELLA®

Java 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.

Information in this document may change without notice. EQUELLA® and its accompanying documentation are furnished under a non-disclosure, evaluation agreement or licence agreement. Copying, storing, transmitting, or otherwise reproducing the software or this document in any form without written permission from Pearson is strictly forbidden.

All products, other than EQUELLA®, named in this document are the property of their respective owners.

Property of:

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

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

Table of Contents

Java SOAP tutorial

Running the examples

To compile and execute the example code you will need to download a number of library jar files. See the readme.txt file in the **samples/java/lib** directory for more information.

Use of an Integration Development Environment (IDE) is recommended for editing of the files. Eclipse is a free IDE available at http://www.eclipse.org

You can create a new project file in Eclipse by selecting **New** -> **Project**... -> **Java Project** and click **Next**. Unselect the Use default location checkbox and browse to the **samples/java** folder of this integration pack. Now click **Finish**.

To run the example files, modify the **settings.properties** in the **samples/java/src/org/example/soap** folder 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 classes in the **com.tle.web.remoting.soap** package were generated with Apache Axis with the following command line:

```
java -cp axis.jar;commons-logging-1.0.4.jar;commons-discovery-0.2.jar;jaxrpc.jar;wsdl4j-1.5.1.jar org.apache.axis.wsdl.WSDL2Java
http://INSTITUTION_URL/services/SoapService50?wsdl
```

In this way, you can build classes to invoke SOAP methods for any SOAP service exposed by EQUELLA. E.g. the copyright SOAP service at *http://INSTITUTION_URL/services/calactivation.service?wsdl*

The example code contains a helper class (EQUELLASOAP) which you are free to use and modify. This class has been created to simplify the processing of the XML results and XML parameters returned from and supplied to the SOAP methods, it also ensures the user's session is maintained between SOAP calls by setting setMaintainSession(true) on the SOAP service locator object. The EQUELLASOAP class is not required to use the EQUELLA SOAP services, it is simply for convenience.

Search

This example shows how you can retrieve a list of items from EQUELLA based on a few search parameters. The searching code is contained within the cmdSearch_Click method. Here is a summary of that code:

- It creates an EQUELLASOAP object which is a thin wrapper around the automatically generated SoapService50 object. The EQUELLASOAP ensures that user session is maintained on the SoapService50 object and logs the user in.

- It invokes searchItems on the EQUELLASOAP object. This method is also a thin wrapper around the SoapService50 method. This method returns an XMLWrapper object for convenience, rather than raw XML string returned by the SoapService50.
- 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. The contribution code is contained within the cmdSubmit_Click method. Here is a summary of that code:

- It creates an EQUELLASOAP object which is a thin wrapper around the automatically generated SoapService50 object. The EQUELLASOAP ensures that user session is maintained on the SoapService50 object and logs the user in.
- It invokes newItem on the EQUELLASOAP object. 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 EQUELLASOAP class converts the binary file into a base 64 encoded string for passing to the SOAP 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 true. A submit value of true means that the item will now either go live or move onto the next moderation step, a value of false 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 SoapService50 object.

When creating a SoapService50 object, you should supply the URL to connect to although this is not strictly necessary. You MUST set a cookie handler on the SoapService50 object before invoking the login method. This is because the user session is retrieved from the cookie by the EQUELLA server; if you do not set a cookie handler your session will be lost immediately after calling login. When using Apache Axis, you can easily do this by setting setMaintainSession(true) on the generated locator class.

2. Call the login method:

```
client.login(username, 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:

```
collectionsXml = client.getContributableCollections();
```

(NOTE: All SOAP methods will return either a simple string, number or an XML string. We recommend using the XmlDocument class 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.