Java SOAP Tutorial

Contents

[Running the examples 2](#_Toc238538868)

[Guide to the example code 2](#_Toc238538869)

[Writing your own code 3](#_Toc238538870)

## Running the examples

The samples/java folder contains an Eclipse IDE project file. Eclipse is a free IDE available at <http://www.eclipse.org>

Import the project file into Eclipse by selecting File -> Import... -> Existing Projects into Workspace and browsing to the samples/java folder.

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/SoapService41

For example: http://mydomain.edu.au/myinst/services/SoapService41

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/SoapService41?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 SoapService41 object. The EQUELLASOAP ensures that user session is maintained on the SoapService41 object and logs the user in.
* It invokes searchItems on the EQUELLASOAP object. This method is also a thin wrapper around the SoapService41 method. This method returns an XMLWrapper object for convenience, rather than raw XML string returned by the SoapService41.
* 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 SoapService41 object. The EQUELLASOAP ensures that user session is maintained on the SoapService41 object and logs the user in.
* It invokes newItem on the EQUELASOAP 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 SoapService41 object**

When creating a SoapService41 object, you MUST also the URL to connect to. The WSDL files generated by the EQUELLA server have an incorrect URL within them.

You MUST also set a cookie handler on the SoapService41 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.

1. **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 Resource Centre.

1. **Invoke a method**

collectionsXml = client.getContributableCollections();

Note that 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.

1. **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.