SRU Client User Guide

Introduction

The SRU client was developed as part of the JISC-funded CD-LOR (Community Dimensions of Learning Object Repositories) project. The purpose of this tool is to promote the contents of a learning object repository such that users will be encouraged to register for the repository and start engaging with it.

The tools will only be able to search repositories that support the standard search protocol called SRU (Search/Retrieve via URL). Specifically, this SRU client has only been tested with intraLibrary (v2.8) but should work with any repository using SRU.

How it works

A search is submitted by an HTML form from a normal web page. This sends an SRU-formatted GET request to the specified SRU server (part of the Learning Object Repository) and a search is issued.

You can search 'all fields' which on intraLibrary will do a 'simple search' but on other repositories the implementation details of an unrestricted search can vary. You can obviously also restrict the search using the drop-down. The values that appear in the drop-down are configured from a JavaScript file (see Implementation section). The field restrictions are standard Dublin Core (DC) fields.

The response from the repository is in XML and the returned XML is transformed into XHTML using the XSL style sheet specified in the original request. Note that this transformation is done entirely by the browser.

There is a slight awkwardness due to the security settings of most browsers. The XSL style sheet will only be applied if it is served from the same server that is hosting the target of the search form. In other words, the XSL file must sit on the same server as your Learning Object Repository.

The results are displayed in XHTML by the Web browser with superficial styling determined by an external CSS (cascading style-sheets) file.

Implementation and Configuration

In order to use the SRU client tool, it needs to be deployed on a running Web server. In other words, you cannot run it from the file system and expect it to work. This is because of security implications when applying styles to retrieved XML docs.

You should ensure your repository is configured such that external searches can see the objects in your repository. In intraLibrary this is defined for each collection and the default is for a collection of objects not to be searchable by external systems. IntraLibrary administrators can change this setting in “collections” area found in “admin tools”. Click “Edit properties” next to the collection you wish to make visible and select “Allow published content in this collection to be searched by external systems”.

The SRU client is stored in the SourceForge Open Source Software Repository at: http://sruclient.sourceforge.net/ From this site, you will be able to download the latest version of the SRU client tool.

The SRU client consists of a number of files most of which we expect you will want to configure for your own users.

index.html
This XHTML file contains the form which the user completes and submits to carry out a search of the learning object repository. This file can sit on any Web server, not necessarily the same server that hosts the learning object repository. The code contained in this page could be embedded into any other web page but you should ensure that the links to the cascading style-sheets *sru.css* and *results.css* are maintained (found in the header of *index.html*).

The form contains three elements. The first is defining the SRU service you wish to search. In cases where you would only ever want to search one repository this element could be removed (see *data.js*). The list of available SRU services is obtained from *data.js* contained in the directory *js*.

The second form element determines which metadata fields will be searched. The default setting is for “All fields” to be searched and this will do an “unrestricted” search on the repository. In the case of intraLibrary this is performing a “simple search” but other repositories will vary in their implementations. The user can decide to restrict his search by selecting from the drop-down menu. The elements found in this drop-down menu are defined in the file *data.js*. Again, you may choose to simplify your form by taking this element away such that every search submitted will be unrestricted (see *data.js*).

The third element of the form is for the search term the user wishes to find objects for. The search functionality is fairly simple and doesn’t support things like Boolean operators (AND OR).

data.js

This JavaScript file can be found in the directory *js* and should exist on the same server as *index.html*. It is in this file that a number of variables associated with the SRU client can be set. Details can be found in the file as commented text but to summarize you can define the following:

- The SRU servers* which appear in the form in *index.html*
- The metadata fields which appear in the second menu in *index.html*
- The URL of the XSL style-sheet required to convert the search results into XHTML (must be in the same domain as the LOR)
- If there is only one SRU server defined you have the option to hide the first form field in *index.html*
- If there are no metadata fields defined you can hide the second form field in *index.html*
- How many results are returned – the default number is 5

*For those using intraLibrary, the URL you need can be derived from the URL of your intraLibrary installation. For example, if your intraLibrary can be found at http://www.location.ac.uk/intraLibrary/index.jsp then the URL of the SRU server for that installation is http://www.location.ac.uk/intraLibrary/IntraLibrary-SRU.

sru.js

This is another JavaScript file which can also be found in the directory *js*. This provides much of the functionality required to submit the search from *index.html* and it is very unlikely that you would want to change this code.

searchretrieve.xsl

This is the XSL style-sheet which defines how the XML returned by the SRU server gets converted to XHTML such that the Web browser can display the results of the search. The XHTML produced is the simplest needed to effective display the search results. The actual aesthetics of the results page are defined entirely through the cascading style-sheet *results.css*. Be sure to define the full URL of *results.css* in *searchretrieve.xsl* (you can find the link within the <head> tag).

Those comfortable with XSL and CSS have a great deal of freedom with regards to structuring the results page so that is looks like other pages in their institutional/project website. Certainly, you will probably want to change the wording to your needs if nothing else. Even those who don’t know XSL can easily find he relevant bits of text to change.

Remember that, due to security issues, the XSL file has to reside on the same server as the learning object repository. If using intraLibrary you could store the XSL file in the library and refer to it using the Public URL functionality (see intraLibrary guidelines) although you would want to configure things so regular users of your library could not find this object.

You can change the information being displayed for each object by adding the appropriate metadata field. You can just copy another metadata field and replace with the relevant reference, in this case dc.type (see comments in *data.js* for full list of references for metadata fields). For example, to add the resource type you need to insert the following code:

```
<xsl:for-each select='purl_dc:type'>
   <div class="metadataField">
        <span class="leftColumnResults">
             <span class="metadataHeading">resource type:</span>
        </span>
        <span class="rightColumnResults">
             <span class="metadataContent">
                  <xsl:value-of select='.' />
             </span>
        </span>
   </div>
</xsl:for-each>
```

The code is already there, but commented out, so all you need to do is remove the comment marks.

sru.css and **results.css**

These cascading style-sheet (CSS) files can be found in the directory *style* and should sit on the same server as *index.html*. Anything to do with the design of the search form or results page (font, color, position, etc.) can be defined here. If you want to do really fancy things you may need to add some new classes and elements in *searchretrieve.xsl*, but the structure of the document (defined in *searchretrieve.xsl*) and style classes (defined here) provided should be enough for general use.