DSpace SWORDv2: A Technology How-To

Richard Jones, Cottage Labs, richard@cottagelabs.com

Abstract: DSpace has had SWORDv2 support since the 1.8 release, and since then usage of that interface has seen some steady but slow take-up. This paper introduces the protocol operations provided by the standard, and how they relate to DSpace. The ways that DSpace can be configured to behave during deposit operations is detailed, so that developers have enough information to build and deploy their own solutions using DSpace's SWORDv2 implementation.

DSpace has had SWORDv2 [1] support since the 1.8 release [2], and since then usage of that interface has seen some steady but slow take-up.

The module which provides this support was written under funding from the SWORDv2 project [3], which completed over a year ago now. Nonetheless, development of the module has continued, using funding from a variety of sources, and in particular the University of Oslo-funded DUO project [4] and the Jisc-funded Portfolio Commons [5] project.

The result of this work is that a new DSpace SWORDv2 module is available separately in github [6], which is considerably more stable than the existing DSpace implementation, and contains more configuration options.

SWORDv2 supports the 4 operations which make up CRUD (Create, Retrieve, Update, Delete), and these are manifested in DSpace thus:

Create - A new DSpace Item can be created in a user workspace by sending either just the metadata or both the metadata and the bitstreams to a DSpace collection's SWORDv2 endpoint. Once the item is created it will either remain in the user workspace or be injected into the workflow, depending on the parameters provided by the depositor. Dublin Core is supported by default as the metadata format, and other formats can be supported by providing the relevant plugin. Any package of bitstreams provided during a create operation is unpacked by a plugin which is loaded based on the format of that package, meaning that the addition of support for other formats is straightforward; furthermore, package plugins for SWORDv2 can utilise the existing DSpace packager plugins.

Retrieve - There are a number of things that can be retrieved from DSpace via SWORDv2: the metadata only, a package of all of the content of an item, or the individual bitstreams themselves. Again, providing symmetry with the create process, Dublin Core is supported by default as the metadata format that is disseminated, while other formats can be supported by providing a plugin. If a package of all of the content of an item is requested, then a package disseminator plugin for that format is invoked; and again, package dissemination plugins for

SWORDv2 can utilise the existing DSpace package disseminators.

Update - Metadata or bitstreams for a DSpace Item can be either replaced or added to via the item's own SWORDv2 endpoint. The exact behaviour (add or replace) is controlled by the HTTP method used by the client. In either case, the metadata and bitstreams are handled by plugins in exactly the same way as in a create operation. In order to deal with the complexities of how updates work, the SWORDv2 module has plugins and configuration which control aspects such as: Can an item be updated in its current state (workspace, workflow, archive, withdrawn)? Which metadata fields can be replaced (there may be some local fields the administrator has added which we do not want overwritten)? Are replaced files totally deleted, or are they backed up somewhere (soft-deleted)?

Delete - DSpace will allow the removal of an entire item, all of the bitstreams associated with an item, and individual bitstreams themselves through the SWORDv2 endpoint. As with update, there are complexities as to how deletes work. In order to handle these issues, there are plugins and configuration which deal with aspects like: Can an item be deleted from or modified in its current state (workspace, workflow, archive, withdrawn)? If a bitstream (or set of bitstreams) is being deleted, are they totally removed, or are they backed up somewhere?

This presentation is not intended as a tutorial, but offers to make the operations of SWORDv2 in DSpace more transparent to potential users/developers, and to provide pointers to resources which will aid implementation, and to expose the key areas that implementers need to think about and know about when implementing their deposit solution.

We will begin by enumerating the SWORDv2 protocol operations laid out above, and how these are interpreted within the context of DSpace. We will then go on to look at each protocol operation in more detail and see how their behaviour can be configured, where plugins to control DSpace behaviour can be applied or modified, and how authentication and authorisation play a role.

We will conclude by briefly considering the future of the SWORDv2 endpoint, both in relation to other DSpace APIs (such as the REST API), and also how in future more package and metadata formats could be supported by the community.

[1] SWORDv2 Profile: http://swordapp.github.com/SWORDv2-Profile/SWORDProfile.html
[2] DSpace 18: https://wiki.duraspace.org/display/DSDOC18/DSpace+1.8+Documentation
[3] SWORDv2 Project: http://swordapp.org
[4] DUO project: http://cottagelabs.com/projects/duo
[5] Portfolio Commons project: http://cottagelabs.com/projects/portfolio-commons
[6] DSpace SWORDv2 enhanced version: https://github.com/swordapp/DSpaceSWORDv2