Using Curl for Direct Service Invocation

Curl (**curl**.haxx.se) is a command line tool useful for invoking web services. In addition to the restful services listed at http://sead.ncsa.illinois.edu/sead-acr-api, SEAD provides a command line tool and session management services to simplify authentication:

Authentication and session management:

To acquire a Google access token, invoke the following tool:

```
java -cp sead1.2b.jar org.sead.acr.client.SEADGoogleLogin
```

(or follow the alternative directions below to use Curl and manage this process yourself)

The first invocation will initiate a Google "device" authorization process as with the SEADUploader: You will see messages such as:

```
Did not find stored refresh token. Initating first-time device authorization request.

1) Go to : http://www.google.com/device in your browser

2) Type : 2c47gdhu in your browser

3) Hit <Return> to continue.
```

Once you've entered the code in a browser and hit <return>, the process will complete with information as in the example below:

```
Proceeding
New Access Token is: ya29.KgD2pVWLsA5TBR4AAACbQLkXOa0nnxD0Es2taQPbx5ebeKR0BJ02VmMJS-7CXg
Expires in 3600 seconds
```

On subsequent invocations, you'll simply see:

```
New Access Token is: ya29.KgAEcQrmKtUJTx4AAABuo3_Qs2iTnlSOp4dpNz9EWCUuSpDcVmGiQZh2YYunsg

Expires in 3600 seconds
```

This access token can then be used to login to SEAD and retrieve a session cookie that can be used for all subsequent service requests:

```
curl --cookie-jar <temporary cookie file> -d "username=<your email>&googleAccessToken=<your access token" http://<desired ACR space>/acr/api/authenticate
```

Calling SEAD services

With a session cookie (generated by the call to /acr/api/authenticate), calling SEAD services is straight forward, For example,

```
curl -b <temporary cookie file> http://<desired ACR space>/acr/resteasy/collections
```

will retrieve JSON Linked Data (JSON-LD - see http://json-ld.org/) formatted metadata for the top-level collections in the specified ACR space:


```
{
    "tag:cet.ncsa.uiuc.edu,2008:/bean/Collection/75D08C13-A727-4631-8401-555CFA330411": {
        "Identifier": "tag:cet.ncsa.uiuc.edu,2008:/bean/Collection/75D08C13-A727-4631-8401-555CFA330411",
        "Title": "el_BOB",
        "Date": "2013-10-01T19:10:14.155Z",
        "Uploaded By": "http://cet.ncsa.uiuc.edu/2007/person/admin",
        "Creator": "Coleman, Eric : http://vivo-vis-test.slis.indiana.edu/vivo/individual/a1375"
    },
    "tag:cet.ncsa.uiuc.edu,2008:/bean/Collection/1339ECC8-6FAE-4C63-BDBE-DA7FC1999965": {
        "Identifier": "tag:cet.ncsa.uiuc.edu,2008:/bean/Collection/1339ECC8-6FAE-4C63-BDBE-DA7FC1999965",
        "Title": "Test",
        "Date": "2014-01-13T18:55:58.082Z",
        "Uploaded By": "http://cet.ncsa.uiuc.edu/2007/person/anonymous"
```

```
    },
    "tag:cet.ncsa.uiuc.edu,2008:/bean/Collection/41c3486d-a18f-468f-a0fa-50c1bb5102a8": {
        "Identifier": "tag:cet.ncsa.uiuc.edu,2008:/bean/Collection/41c3486d-a18f-468f-a0fa-50c1bb5102a8",
        "Title": "test-classes",
        "Date": "2014-03-27T16:25:37.536Z",
        "Uploaded By": "http://cet.ncsa.uiuc.edu/2007/person/admin",
        "Abstract": "cheese"
    },
    "tag:medici@uiuc.edu,2009:col_BJtxI2-d8-ng5TpQPNcdlA": {
        "Identifier": "tag:medici@uiuc.edu,2009:col_BJtxI2-d8-ng5TpQPNcdlA",
        "Title": "classes",
        "Date": "2014-05-28T13:36:42.543Z",
        "Uploaded By": "http://cet.ncsa.uiuc.edu/2007/person/myersjd@umich.edu"
    },
    "@context": {
        "Identifier": "http://purl.org/dc/elements/1.1/identifier",
        "Title": "http://purl.org/dc/elements/1.1/title",
        "Date": "http://purl.org/dc/terms/created",
        "Uploaded By": "http://purl.org/dc/elements/1.1/creator",
        "Abstract": "http://purl.org/dc/terms/abstract",
        "Contact": "http://sead-data.net/terms/contact",
        "Creator": "http://purl.org/dc/terms/creator"
    }
}
```


Note that in JSON-LD, the @context can be used to associate the labels given with RDF identifiers for the predicates.

The identifiers of the collections can be used to get further information about them:

curl -b <temporary cookie file> http://<desired ACR space>/acr/resteasy/collections/<id> - the same metadata as above for one collection

curl -b <temporary cookie file> http://<desired ACR space>/acr/resteasy/collections/<id>/biblio - further bibliographic metadata

curl -b <temporary cookie file> http://<desired ACR space>/acr/resteasy/collections/<id> - all known metadata (biblio, user-provided, and extracted metadata)

curl -b <temporary cookie file> http://<desired ACR space>/acr/resteasy/collections/<id>/datasets - the list of datesets in the collection

curl -b <temporary cookie file> http://<desired ACR space>/acr/resteasy/collections/<id>/collections - the list of subcollections

Curl can be used to create collections, upload data (and provide provenance/metadata in the same call), add tags, add metadata, etc.

For example, uploading a dataset would be done like this:

curl -v -b \tmp\temp.txt --form "datablob=@readme.txt" --form "http://purl.org/vocab/frbr/core#embodimentOf=/test/readme.txt" http:/<your ACR space/acr/resteasy/datasets

which would upload a readme.txt file and add metadata identifying the path for it on the local disk.

Downloading would be done with:

curl -b <temporary cookie file> -O http://<your ACR space>/acr/resteasy/datasets/{id}/file

Adding metadata to an existing dataset would be done as:

curl -v -b \tmp\temp.txt --form "http://sead-data.net/terms/sbdc/ph=6.64;type=text/plain" --form "http://purl.org/dc/terms/isPartOf=http://dx.doi.org/10.1175/BAMS-D-11-00180.1;type=text/uri-list" http:/<your ACR space/acr/resteasy/datasets/<id>/metadata

with the first setting a text value (for pH) and the second setting a URI (i.e. as an external identifier for a dataset that this one is part of).

Collections have an analogous service - simply swap "collections" for "datasets" in the URL used above to set metadata for a collection.

To set a relationship, simply use the identifier of the dataset/collection for the object with type=text/uri-list. For any predicate that has been configured as a relationship, this will automatically create the relation and the reverse relationship.

Curl Tips:

cookie files and the destination for downloaded files need to be in writable directories

-v (verbose) is a useful flag to see more about the service request/response

Using Curl for Google login:

Curl can be used instead of SEAD's SEADGoogleLogin utility to initially register a device (your computer) with Google and acquire SEAD credentials as well as to acquire new tokens over time (see https://developers.google.com/accounts/docs/OAuth2ForDevices for details):

curl -v -d "client_id=972225704837-hlsr4m5f69v2s0vig8c6fbdmdohvqjv6.apps.googleusercontent.com&scope=email profile" https://accounts.google.com/o/oauth2/device/code

will request a user_code that can be entered in a browser.

curl -v -d "client_id=972225704837-hlsr4m5f69v2s0vig8c6fbdmdohvqjv6.apps.googleusercontent.com&client_secret=<SEAD client secret>&code=<device _code from step 1>&grant_type=http://oauth.net/grant_type/device/1.0" https://accounts.google.com/o/oauth2/token

will acquire an access token (used by SEAD) and a refresh_token (used to acquire new access_tokens in the future). Your code should store the refresh_token for future use to avoid having your user deal with the code step.

curl -v -d "client_id=972225704837-hlsr4m5f69v2s0vig8c6fbdmdohvqjv6.apps.googleusercontent.com&client_secret=<SEAD Client Secret>&refresh_token=<refresh_token from step 2>&grant_type=refresh_token" https://accounts.google.com/o/oauth2/token

will acquire an access token from the refresh token.

As with the java based method, this access token can then be used to login to SEAD and retrieve a session cookie that can be used for all subsequent service requests:

curl --cookie-jar <temporary cookie file> -d "username=<your email>&googleAccessToken=<your access token" http://<desired ACR space>/acr/api/authenticate