

DVN API for Sharing Data

The URLs for the DVN API resources are of the form

```
/dvn/api/<Verb Name>{/arg}{?{{arg}}&...}}
```

Generally, mandatory arguments are embedded in the URL and optional arguments are supplied as query parameters, in the `?param=...` notation. See the documentation for the individual resources below for details.

The API supports basic HTTP Authentication. So that the access credentials are not transmitted in the clear,
the API verbs (methods) below are **only accessible over HTTPS**.

I. Metadata API

The API for accessing DVN metadata is implemented in 4 verbs (resources):

```
metadataSearchFields
metadataSearch
metadataFormatsAvailable
metadata
```

1. metadataSearchFields

Arguments:

none

URL example:

```
/dvn/api/metadataSearchFields/
```

Output:

XML record in the format below:

```
<MetadataSearchFields>
<SearchableField>
<fieldName>title</fieldName>
<fieldDescription>title</fieldDescription>
```

```

</SearchableField>
<SearchableField>
<fieldName>authorName</fieldName>
<fieldDescription>authorName</fieldDescription>
</SearchableField>
<SearchableField>
<fieldName>otherId</fieldName>
<fieldDescription>otherId</fieldDescription>
</SearchableField>
...
</MetadataSearchableFields>

```

2. metadataSearch

Arguments:

queryString: mandatory, embedded.

Standard Lucene-style search queries are supported; (same query format currently used to define OAI sets, etc.)

URLs examples:

```

/dvn/api/metadataSearch/title:test
/dvn/api/metadataSearch/title:test AND authorName:leonid

```

Output:

XML record in the format below:

```

<MetadataSearchResults>
<searchQuery>title:test</searchQuery>
<searchHits>
<study ID="hdl:TEST/10007"/>
...
</searchHits>
</MetadataSearchResults>

```

Error Conditions:

Note that when the query does not produce any results, the resource returns an XML record with an empty <searchHits> list, NOT a 404.

3. metadataFormatsAvailable

Arguments:

objectId: mandatory, embedded.

Both global and local (database) IDs are supported.

URLs examples:

```

/dvn/api/metadataFormatsAvailable/hdl:1902.1/6635

```

/dvn/api/metadataFormatsAvailable/9956

Output:

XML record in the format below:

```
<MetadataFormatsAvailable studyId="hdl:TEST/10007">

<formatAvailable selectSupported="true" excludeSupported="true">
<formatName>ddi</formatName>
<formatSchema>http://www.icpsr.umich.edu/DDI/Version2-0.xsd</formatSchema>
<formatMime>application/xml</formatMime>
</formatAvailable>

<formatAvailable>
<formatName>oai_dc</formatName>
<formatSchema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd</formatSchema>
<formatMime>application/xml</formatMime>
</formatAvailable>

</MetadataFormatsAvailable>
```

(note the selectSupported and excludeSupported attributes above!)

Error Conditions:

404 NOT FOUND
if study does not exist

4. metadata

Arguments:

objectId: mandatory, embedded.
Both global and local (database) IDs are supported.

formatType: optional, query.
Defaults to ddi if not supplied.

URLs examples:

```
/dvn/api/metadata/hdl:1902.1/6635
/dvn/api/metadata/9956
/dvn/api/metadata/hdl:1902.1/6635?formatType=ddi
```

Output:

Metadata record in the format requested, if available.
No extra headers, etc.

Partial selection of metadata sections:

When requesting partial records is supported (see `metadataFormatsAvailable`, above for more info), these additional parameters can be supplied:

partialExclude: optional, query.

Xpath query representing metadata section to drop, where supported.

partialInclude: optional, query.

Xpath query representing metadata section to include, where supported.

Examples:

`/dvn/api/metadata/hdl:1902.1/6635?formatType=ddi&partialExclude=codeBook/dataDscr`
will produce a DDI without the dataDscr section. *[I'm expecting this to be the single most useful and common real-life application of this feature - L.A.]*

`/dvn/api/metadata/hdl:1902.1/6635?formatType=ddi&partialInclude=codeBook/stdyDscr`
will produce a DDI with the stdyDscr section only.

(for now, only simple top-level Xpath queries like the above are supported).

One other limitation of the current implementation: it does not validate the supplied `partialExclude` and `partialInclude` arguments; no error messages/diagnostics will be given if the Xpath queries are not part of the metadata schema.

For example, if you request `partialInclude=foobar`, it will quietly produce an empty DDI, and `partialExclude=foobar` will not exclude anything (and you will get a complete DDI).

Error Conditions:

404 NOT FOUND
if study does not exist

503 SERVICE UNAVAILABLE
if study exists, but the format requested is not available;
also, when partial exclude or include is requested, if it's not supported by the service (see the documentation for `metadataFormatsAvailable` above).

Notes:

A real-life workflow scenario may go as follows:

- Find the searchable index fields on this DVN (`meatadataSearchFields`)
- Run a search (`metadataSearch`)
- For [select] studies returned, find what metadata formats are available (`metadataFormatsAvailable`)
- Retreive the metadata in the desired format (`metadata`)

II. File Access API

The DVN API for downloading digital objects (files) is implemented in 2 verbs (resources):

downloadInfo
download

1. downloadInfo

Arguments:

objectId: mandatory, embedded.
Database ID of the DVN Study File.

URLs example:

/dvn/api/downloadInfo/9956

Output:

XML record in the format below:

(Note: the record below is only an example; we will provide full schema/documentation of the FileDownloadInfo record format below)

```
<FileDownloadInfo>
<studyFile fileId="9956">

<fileName>prettypicture.jpg</fileName>
<fileMimeType>image/jpeg</fileMimeType>
<fileSize>52825</fileSize>

<Authentication>
  <authUser>testUser</authUser>
  <authMethod>password</authMethod>
</Authentication>

<Authorization directAccess="true"/>

<accessPermissions accessGranted="true">
  Authorized Access only
</accessPermissions>

<accessRestrictions accessGranted="true">
  Terms of Use
</accessRestrictions>

<accessServicesSupported>

  <accessService>
    <serviceName>thumbnail</serviceName>
    <serviceArgs>imageThumb=true</serviceArgs>
```

```
<contentType>image/png</contentType>
<serviceDesc>Image Thumbnail</serviceDesc>
</accessService>

</accessServicesSupported>
</studyFile>
</FileDownloadInfo>
```

Error Conditions:

404 NOT FOUND
Study file does not exist.

2. download

Arguments:

objectId: mandatory, embedded.
Database ID of the DVN Study File.

Optional Query args:
As specified in the output of downloadInfo, above.

URLs examples:

```
/dvn/api/download/9956
/dvn/api/download/9956?imageThumb=true
/dvn/api/download/9957?fileFormat=stata
```

Output:

Byte Stream
(with proper HTTP headers specifying the content type, file name and such)

Error Conditions:

404 NOT FOUND
Study file does not exist.

401 AUTHORIZATION REQUIRED
Access to restricted object attempted without HTTP Authorization header supplied.

403 PERMISSION DENIED
HTTP Authorization header supplied, but the authenticated user is not authorized to directly access the object protected by Access Permissions and/or Access Restrictions ("Terms of Use").