Package 'dvn'

July 2, 2014

Version 0.3.3

Date 2013-11-29

Title Access to The Dataverse Network APIs

Author Thomas J. Leeper

Maintainer Thomas J. Leeper <thosjleeper@gmail.com>

Imports RCurl, XML

Description Provides access to The Dataverse Network APIs via the RCurl package, enabling access to archived data (and metadata), and the ability to create and manipulate studies in a user's dataverse(s).

License GPL-2

URL https://github.com/leeper/dvn

BugReports https://github.com/leeper/dvn/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-02 21:50:19

R topics documented:

---

dvn-package *Access to The Dataverse Network APIs*

Description

Provides access to The Dataverse Network Data Sharing and Data Deposit APIs via RCurl.

The Data Sharing API provides access to archived data (and metadata). The dvn package provides a number of functions to translate API responses into R data structures using the XML package.

The Data Deposit API provides the ability to create and modify Dataverse studies. Support for these features is in ongoing development.

Details

Package:	dvn
Type:	Package
Version:	0.3.3
Date:	2013-11-29
License:	GPL-2

Author(s)

Thomas J. Leeper

Maintainer: Thomas J. Leeper <thosjleeper@gmail.com>

References

Dataverse Data Sharing API Documentation

Dataverse Data Deposit API Developer's Guide

Dataverse Data Deposit API v1 Developer's Guide

The Dataverse Network

Harvard IQSS Dataverse

Examples

```
## Not run:
    ## A basic Data Sharing API workflow:
    # find available search fields
    fields <- dvSearchFields()

    # search within those fields
    search <- dvSearch(list(authorName="leeper"))

    # retrieve available metadata formats for a search result
    formats <- dvMetadataFormats(search$objectId[1])

    # retrieve metadata in an available format and find fileId(s)
   files <- dvExtractFileIds(dvMetadata(search$objectId[1], format.type=formats$formatName[1]))

    # optional search for information about data download
    info <- dvDownloadInfo(files$fileId[1])

    # download the data, if allowed by Terms of Use
    data <- dvDownload(files$fileId[1])

    # change Dataverse network (to Odum Institute)
    options(dvn='https://arc.irss.unc.edu/dvn/')
    search <- dvSearch(list(authorName="leeper"))

## End(Not run)
## Not run:
    ## Use basic Data Deposit API functions:
    options(dvn.user="username")
    options(dvn.pwd="password")
    # retrieve my Dataverses and their terms of use
    dvServiceDoc()

    # Retrieve available studies in a user's dataverse with either:
    user <- dvUserStudies("mydataverse")
    # Or:
    user <- dvUserStudies(dvServiceDoc())

    # retrieve a 'Study Atom', providing the citation for a study
    dvStudyAtom(user$studies$objectId)

    # retrieve a 'Study Statement' providing files stored in a study
    dvStudyStatement(user$studies$objectId)

## End(Not run)
## Not run:
```

```
## Create a study, add files, and release:
options(dvn.user="username")
options(dvn.pwd="password")
# Check credentials and accessible dataverses
dvServiceDoc()

# Create study metadata and create study with it
writeLines(dvBuildMetadata(title="My Study"),"mystudy.xml")
created <- dvCreateStudy("mydataverse","mystudy.xml")

# Get `objectId` from `dvUserStudies`
objectid <- dvUserStudies("mydataverse")$studies$objectId[1]
# Or, retrieve dataverse `objectid` from `dvCreateStudy` receipt
objectid <- created$objectId

# Add files and release study using `objectid`
dvAddFile(objectid,"mydata.zip")
# or add multiple files:
dvAddFile(objectid,c("file1.csv","file2.txt"))
# Release study
dvReleaseStudy(objectid)

## End(Not run)
```

`dvAddFile` *Add a study file*

Description

Use Data Deposit API to add a file to a study.

Usage

```
dvAddFile(  objectid, filename=NULL, dataframe=NULL, dv=getOption('dvn'),
            user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
            browser=FALSE, ...)
```

Arguments

`objectid`	A character string containing the id for a dataverse study (a handle). This could be returned from `dvUserStudies`, `dvCreateStudy`, (or, `dvSearch`, from the Data Sharing API).
`filename`	The path to a .zip file to be uploaded, or a vector of filenames (or paths) that will be compressed then uploaded.
`dataframe`	A character string (or vector of character strings) containing the names of dataframe(s) to be saved and uploaded. Files are named by their R object names, so a dataframe called `df` will become "df.RData".

dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is `options('dvn.user')`.
pwd	A character string containing the corresponding dataverse password. Default is `options('dvn.pwd')`.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to `curlPerform` via `dvDepositQuery`.

Details

This send a (.zip) file to a study, decompresses it, and adds the file contents to the study. To send multiple files, specify them as a character vector in `filename`. It is also possible to add one or more dataframes (in addition to or instead) of filenames.

The user should receive (immediately) an email from Dataverse confirming whether or not decompression of the .zip was successful. Successful file upload can also be checked with a call to `dvStudyStatement`.

Note: Filenames in the .zip must be unique and cannot duplicate the names of any files already attached to the study. Use `dvDeleteFile` to remove previous files and then replace by uploading a new .zip.

Value

If successful, a list containing:

bibliographicCitation	A full citation for the study, compiled from metadata.
generator	A vector containing details about the SWORD application.
id	The full URI for the study.
objectId	The `objectId` for the study.
xml	A character string containing an XML data structure that includes a study citation and handle.

Author(s)

Thomas J. Leeper

See Also

`dvDeleteFile` `dvCreateStudy` `dvEditStudy` `dvReleaseStudy`

`dvBuildMetadata` *Build the Atom object to create or edit a study.*

Description

Build the Qualified Dublin Core Atom XML object needed for `dvCreateStudy` or `dvEditStudy`.

Usage

```
dvBuildMetadata(..., format='dcterms')
```

Arguments

`...`	One or more named metadata fields. "title" is required. Any other fields are optional, but must be in Qualified Dublin Core.
`format`	A character specifying the format of the metadata XML. Default is "dcterms".

Value

A character string of class "dvMetadata" containing a complete Atom entry for use in `dvCreateStudy` or `dvEditStudy`.

Allowed fields are: "abstract", "accessRights", "accrualMethod", "accrualPeriodicity", "accrualPolicy", "alternative", "audience", "available", "bibliographicCitation", "conformsTo", "contributor", "coverage", "created", "creator", "date", "dateAccepted", "dateCopyrighted", "dateSubmitted", "description", "educationLevel", "extent", "format", "hasFormat", "hasPart", "hasVersion", "identifier", "instructionalMethod", "isFormatOf", "isPartOf", "isReferencedBy", "isReplacedBy", "isRequiredBy", "issued", "isVersionOf", "language", "license", "mediator", "medium", "modified", "provenance", "publisher", "references", "relation", "replaces", "requires", "rights", "rightsHolder", "source", "spatial", "subject", "tableOfContents", "temporal", "title", "type", and "valid".

Author(s)

Thomas J. Leeper

References

Dublin Core Metadata Terms

Qualified Dublin Core (Wikipedia)

Atom entry specification

Qualified Dublin Core XML example 1

Qualified Dublin Core XML example 2

See Also

`dvCreateStudy` `dvEditStudy`

Examples

```
## Not run:
dvBuildMetadata(title="My Study",
                creator="Doe, John",
                creator="Doe, Jane",
                publisher="My University",
                date="2013-09-22",
                description="An example study",
                subject="Study",
                subject="Dataverse",
                subject="Other",
                coverage="United States")

## End(Not run)
```

---

dvCreateStudy *Create a study in a named Dataverse*

Description

Use Data Deposit API to create a study in a Dataverse.

Usage

```
dvCreateStudy(  dvname, xmlfile, dv=getOption('dvn'),
                user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
                browser=FALSE, ...)
```

Arguments

dvname	A character string containing the name of a dataverse collection. Must be a dataverse accessible to this user (perhaps as extracted from dvServiceDoc.
xmlfile	The path to an XML file (or a character string of that XML) containing a Qualified Dublin Core Atom/XML document, which specifies the metadata necessary to create the study. See references for an example and dvBuildMetadata.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to curlPerform via dvDepositQuery.

Details

This function creates a study in a dataverse using a Qualified Dublin Core Atom/XML document (possibly built by dvBuildMetadata). This document must contain a title, with all other fields being optional.

For the dvname argument, it is also possible to simply pass the "dvServiceDoc" object directly in lieu of naming a dataverse: dCreateStudy(dvServiceDoc(),"atom.xml"). In this case, if the user has access to more than one dataverse collection, an error will be produced.

Creating a study only changes the study in "draft". To make a study public (and thus visible online or via dvSearch), the study needs to be subsequently released with dvReleaseStudy. It can also be deleted with dvDeleteStudy. Note: A released study can be "deaccessioned" (replacing its public record with a publicly visible record about its previous release) using dvDeleteStudy but cannot be deleted.

Value

If successful, a list containing:

bibliographicCitation	A full citation for the study, compiled from metadata.
generator	A vector containing details about the SWORD application.
id	The full URI for the study.
objectId	The objectId for the study.
xml	A character string containing an XML data structure that includes a study citation and handle.

Author(s)

Thomas J. Leeper

References

Atom entry specification

Qualified Dublin Core XML example 1

Qualified Dublin Core XML example 2

See Also

dvBuildMetadata dvAddFile dvEditStudy dvReleaseStudy dvDeleteStudy

dvDeleteFile *Delete a study file*

Description

Use Data Deposit API to delete a study file.

Usage

```
dvDeleteFile(   fileid, dv=getOption('dvn'),
                user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
                browser=FALSE, ...)
```

Arguments

fileid	A character string specifying a dataverse fileId for a file contained in a dataverse record. This is distinct from the objectid that identifies the entire study in which the file is located. This could be returned from dvCreateStudy, dvStudyStatement, (or, dvExtractFileIds, from the Data Sharing API).
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to curlPerform via dvDepositQuery.

Details

This deletes a file in a study.

Value

If successful, the response is an empty character string.

Author(s)

Thomas J. Leeper

See Also

dvAddFile dvCreateStudy dvEditStudy dvReleaseStudy

dvDeleteStudy *Delete (or deaccession) a study*

Description

Use Data Deposit API to delete an unreleased study or deaccession a released study.

Usage

```
dvDeleteStudy( objectid, dv=getOption('dvn'),
               user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
               browser=FALSE, ...)
```

Arguments

objectid	A character string containing the id for a dataverse study (a handle). This could be returned from dvUserStudies, dvCreateStudy, (or, dvSearch, from the Data Sharing API).
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvDepositQuery.

Details

If a study has been created but not released, dvDeleteStudy will delete it permanently.

If a study has been created and released, dvDeleteStudy will deaccession it (i.e., reverse the release). A released study can be "deaccessioned" (replacing its public record with a publicly visible record about its previous release) using dvDeleteStudy. Currently, a previously released study cannot be deleted. Attempting to delete a deaccessioned study will produce an error.

Value

If successful, the response is an empty character string.

Author(s)

Thomas J. Leeper

dvDepositQuery *Execute a Data Deposit API Query*

Description

Primarily an internal function for executing Data Deposit API calls.

Usage

```
dvDepositQuery( query, fulluri=NULL, dv=getOption('dvn'),
                user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
                browser=FALSE, apiversion='v1', httpverb='GET', ...)
```

Arguments

query	A character string specifying query parameters.
fulluri	Instead of specifying query, dv, and apiversion, one can specify the full URI for a Data Deposit action, perhaps as returned by dvStudyAtom.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
apiversion	A character string specifying the Data Deposit API version. Currently only "v1" (the default).
httpverb	A character string specifying one of "GET" (the default), "POST", "PUT", or "DELETE". Not all verbs work on all SWORD URIs. See Dataverse Developer's Guide.
...	Optionally, additional arguments passed to curlPerform.

Details

Primarily for internal use.

Value

A character string containing an XML data structure.

Author(s)

Thomas J. Leeper

dvDownload *Download a file*

Description

Download a dataverse file into R or in a web browser

Usage

```
dvDownload(fileid, query = NULL, dv = getOption('dvn'), browser = FALSE, ...)
```

Arguments

fileid	A character string specifying a dataverse fileId for a file contained in a dataverse reocrd. This is distinct from the objectid that identifies the entire dataverse records.
query	An optional character string or named list of additional query parameters from dvDownloadInfo.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvQuery.

Details

Due to access restrictions placed on some files, not all files are directly downloadable into R. dvDownload automatically checks whether files can be downloaded directly with dvDownloadInfo and stops if direct access is unavailable. Otherwise the file is downloaded and returned.

Value

The requested file as an R object.

Author(s)

Thomas J. Leeper

See Also

dvDownloadInfo

Examples

```
## Not run:
dvDownload("9956")

## End(Not run)
```

dvDownloadInfo *Retrieve file download information*

Description

Retrieve download information for a dataverse file.

Usage

```
dvDownloadInfo(fileid, dv = getOption('dvn'), browser = FALSE, ...)
```

Arguments

fileid	A character string specifying a dataverse fileId for a file contained in a dataverse record. This is distinct from the objectid that identifies the entire dataverse records.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvQuery.

Value

A list containing details of the (potentially downloadable) file:

fileId	A character string containing the value of "fileid" from the original request.
fileName	A character string containing the name of the requested file.
fileMimeType	A character string containing the MIME type of the requested file.
fileSize	A character string containing the file size in bytes.
authMethod	A character string specifying the authentication method for the request
directAccess	A character string containing a logical for whether the file can be downloaded directly via the API.
accessRestrictions	A character string containing details on any file access restrictions.
accessServicesSupported	A dataframe detailing the available file formats for download.
xml	A character string containing the original API response in XML.

Note: access restrictions on files may prevent them from being downloaded directly in R.

Author(s)

Thomas J. Leeper

See Also

dvDownload

Examples

```
## Not run:
info <- dvDownloadInfo("9956")

## End(Not run)
```

---

dvEditStudy *Edit a study*

Description

Use Data Deposit API to replace a study's metadata.

Usage

```
dvEditStudy(objectid, xmlfile, dv=getOption('dvn'),
            user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
            browser=FALSE, ...)
```

Arguments

objectid	A character string containing the id for a dataverse study (a handle). This could be returned from dvUserStudies, dvCreateStudy, (or, dvSearch, from the Data Sharing API).
xmlfile	The path to an XML file (or a character string of that XML) containing a Qualified Dublin Core Atom/XML document, which specifies the metadata necessary to create the study. See references for an example and dvBuildMetadata.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to curlPerform via dvDepositQuery.

Details

This function replaces a study's metadata. That is, it is not possible to edit some fields of a study's metadata. The SWORD implementation only allows an entire metadata record to be replaced. Thus, if one only wants to edit one field, a complete Qualified Dublin Core XML document. Probably, it is best to store this locally to simply making any future changes.

Note: Editing a study only changes the study in "draft". To make those changes public, the study needs to be subsequently released. A released study can be "deaccessioned" (replacing its public record with a publicly visible record about its previous release) using `dvDeleteStudy` but cannot be deleted.

Value

If successful, a list containing:

`bibliographicCitation`
: A full citation for the study, compiled from metadata.

`generator`
: A vector containing details about the SWORD application.

`id`
: The full URI for the study.

`objectId`
: The `objectId` for the study.

`xml`
: A character string containing an XML data structure that includes a study citation and handle.

If successful, the request will reflect the contents of `xmlfile`.

Author(s)

Thomas J. Leeper

References

Atom entry specification

Qualified Dublin Core XML example 1

Qualified Dublin Core XML example 2

See Also

`dvBuildMetadata` `dvAddFile` `dvCreateStudy` `dvReleaseStudy` `dvDeleteStudy`

dvExtractFileIds *Extract fileId(s) from metadata*

Description

Extract fileId(s) from metadata for a dataverse record.

Usage

```
dvExtractFileIds(xml)
```

Arguments

xml	A character string containing an XML data structure returned by dvMetadata.

Details

Extracts all of the fileIds associated with a retrieved dataverse record. Metadata format must be DDI. The Dublin Core metadata record does not include fileIds.

Value

A dataframe containing fileIds and associated descriptive information for files stored in the dataverse record.

Author(s)

Thomas J. Leeper

See Also

dvMetadata

Examples

```
## Not run:
xml <- dvMetadata("hdl:1902.1/17218")
dvExtractFileIds(xml)

## End(Not run)
```

dvMetadata *Retrieve metadata*

Description

Retrieve metadata for a dataverse record, using its object ID.

Usage

```
dvMetadata(objectid, format.type = NULL, include = NULL, exclude = NULL,
dv = getOption('dvn'), browser = FALSE, ...)
```

Arguments

objectid	A dataverse object ID.
format.type	An optional character string specifying the format of the returned metadata. Available options can be retrieved with dvMetadataFormats. If NULL, "ddi" is used by default.
include	An optional character string specifying what metadata to include in the response.
exclude	An optional character string specifying what metadata to exclude from the response.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvQuery.

Value

An character string of class "dvMetadata" containing an XML data structure of metadata for the specified dataverse record. The exact form and contents of this response depend on the format.type and any include/exclude statements.

Author(s)

Thomas J. Leeper

See Also

dvMetadataFormats dvExtractFileIds dvTermsOfUse

Examples

```
## Not run:
# To retrieve all metadata:
xml <- dvMetadata("hdl:1902.1/17218")
# To retrieve metdata in Dublin Core format (instead of the default ("ddi"):
xml <- dvMetadata("hdl:1902.1/17218", format.type="oai_dc")

# To extract just the study description, without file information:
xml <- dvMetadata("hdl:1902.1/17218", include="codeBook/stdyDscr")
# To extract just basic file description(s), without other study details:
xml <- dvMetadata("hdl:1902.1/17218", include="codeBook/fileDscr")
# To extract just data summaries for all variables, without other study details:
xml <- dvMetadata("hdl:1902.1/17218", include="codeBook/dataDscr")

## End(Not run)
```

---

dvMetadataFormats *Retrieve available metadata formats*

Description

Retrieve the available metadata formats that can be returned for a particular dataverse record.

Usage

```
dvMetadataFormats(objectid, dv = getOption('dvn'), browser = FALSE, ...)
```

Arguments

objectid	A dataverse object ID.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvQuery.

Value

A dataframe containing names and supplemental information for the metadata format(s) available for specified dataverse object.

Author(s)

Thomas J. Leeper

Examples

```
## Not run:
formats <- dvMetadataFormats("hdl:1902.1/17218")

## End(Not run)
```

---

dvQuery *Execute a Data Sharing API Query*

Description

Primarily an internal function for executing Data Sharing API calls.

Usage

```
dvQuery(verb, query = NULL, dv = getOption('dvn'), browser = FALSE, ...)
```

Arguments

verb	An API verb of metadataSearchFields, metadataSearch, metadataFormatsAvailable, metadata, downloadInfo, download.
query	An optional character string specifying additional query parameters.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvQuery.

Details

Primarily for internal use.

Value

A character string containing an XML data structure.

Author(s)

Thomas J. Leeper

dvReleaseStudy *Publicly release a study*

Description

Use Data Deposit API to release a created study.

Usage

```
dvReleaseStudy( objectid, dv=getOption('dvn'),
                user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
                browser=FALSE, ...)
```

Arguments

objectid	A character string containing the id for a dataverse study (a handle). This could be returned from dvUserStudies, dvCreateStudy, (or, dvSearch, from the Data Sharing API).
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to curlPerform via dvQuery.

Value

If successful, a list containing:

bibliographicCitation	A full citation for the study, compiled from metadata.
generator	A vector containing details about the SWORD application.
id	The full URI for the study.
objectId	The objectId for the study.
xml	A character string containing an XML data structure that includes a study citation and handle.

Author(s)

Thomas J. Leeper

See Also

dvCreateStudy dvEditStudy dvDeleteStudy

dvSearch *Search a Dataverse*

Description

Search a Dataverse by keyword(s) in one or more metadata fields.

Usage

```
dvSearch(query, boolean = "AND", dv = getOption('dvn'), browser = FALSE, ...)
```

Arguments

query	Either a list of named search parameters or an unnamed, one-element list containing an appropriately formatted character string of search parameters. Search parameters can be retrieved via dvSearchFields. If specified as a single character string, the search is performed on all available fields using a boolean OR logic.
boolean	A character string specifying the boolean "AND" (the default) or "OR" by which multiple values in the query parameter should be combined. Ignored if query is a single character string or one-element list.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvQuery.

Details

Unlike searching on the dataverse website, the API requires searching within named metdata fields. The set of available metadata fields can be retrieved from dvSearchFields.

Value

A dataframe containing one variable (objectId), made up of character strings containing dataverse objectIds.

Author(s)

Thomas J. Leeper

See Also

dvSearchFields

Examples

```
## Not run:
search1 <- dvSearch(list("authorName:leeper"))
search2 <- dvSearch(list(title="Denmark",title="Sweden"),boolean="OR")
search3 <- dvSearch("puppies")

## End(Not run)
```

`dvSearchFields` *Lookup available search fields*

Description

Lookup names and descriptions of search fields available via the API

Usage

```
dvSearchFields(dv = getOption('dvn'), browser = FALSE, ...)
```

Arguments

`dv`	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
`browser`	A logical specifying whether the query should be executed in a web browser. Default is `FALSE`.
`...`	Optionally, additional arguments passed to `getURL` via `dvQuery`.

Value

A dataframe listing the names (and, sometimes, descriptions) of available search fields for use in `dvSearch` for the target dataverse.

Author(s)

Thomas J. Leeper

See Also

`dvSearch`

Examples

```
## Not run:
fields <- dvSearchFields()

## End(Not run)
```

dvServiceDoc *Get Dataverse Service Document*

Description

Obtain a Data Deposit API Service Document

Usage

```
dvServiceDoc(   dv=getOption('dvn'),
                user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
                browser=FALSE, ...)
```

Arguments

dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is `options('dvn.user')`.
pwd	A character string containing the corresponding dataverse password. Default is `options('dvn.pwd')`.
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvDepositQuery.

Value

A character string of class "dvServiceDoc" containing an XML data structure, which describes the dataverses available to the user and the terms of use for those dataverses.

Author(s)

Thomas J. Leeper

See Also

dvTermsOfUse

dvStudyAtom *Obtain a study's Atom document*

Description

Use Data Deposit API to obtain an Atom document for a study.

Usage

```
dvStudyAtom(objectid, dv=getOption('dvn'),
            user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
            browser=FALSE, ...)
```

Arguments

objectid	A character string containing the id for a dataverse study (a handle). This could be returned from dvUserStudies (or, dvSearch, from the Data Sharing API).
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvDepositQuery.

Value

A list containing:

bibliographicCitation	A full citation for the study, compiled from metadata.
generator	A vector containing details about the SWORD application.
id	The full URI for the study.
objectId	The objectId for the study.
xml	A character string containing an XML data structure that includes a study citation and handle.

Author(s)

Thomas J. Leeper

See Also

dvUserStudies dvStudyStatement

dvStudyStatement *Obtain a study statement*

Description

Use Data Deposit API to obtain a listing of study contents.

Usage

```
dvStudyStatement(   objectid, dv=getOption('dvn'),
                    user=getOption('dvn.user'), pwd=getOption('dvn.pwd'),
                    browser=FALSE, ...)
```

Arguments

objectid	A character string containing the id for a dataverse study (a handle). This could be returned from dvUserStudies (or, dvSearch, from the Data Sharing API).
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvDepositQuery.

Value

A list containing information about the study listing, including:

id	The Data Deposit API URI for the study.
title	The title of the study.
author	The study author.
updated	A timestamp.
locked	A logical as a character string.
latestVersionState	A character string containing, generally, "DRAFT" or "RELEASED".
files	A dataframe containing of files stored in the study collection (and their associated fileId)
xml	A character string containing the original XML response data structure.

Author(s)

Thomas J. Leeper

See Also

dvUserStudies

---

dvTermsOfUse *Terms of Use*

Description

Extract and view Dataverse Terms of Use.

Usage

```
dvTermsOfUse(xml)
```

Arguments

xml	A character string containing an XML data structure returned by dvMetadata or dvServiceDoc.

Details

Extracts the Terms of Use from a "dvServiceDoc" or "dvMetadata" (in DDI format) object, saves that information as temporary HTML file (or files, if dvServiceDoc returns multiple dataverses) and opens the file(s) using browseURL. Temporary files are then immediately deleted.

Value

The value of xml, invisibly.

Author(s)

Thomas J. Leeper

See Also

dvMetadata dvServiceDoc

Examples

```
## Not run:
# Extract TOU from Data Sharing reseponse
# (This only applies when `dvMetadata` is in DDI format)
dvTermsOfUse(dvMetadata("hdl:1902.1/17864"))

# Extract TOU from Data Deposit reseponse:
dvTermsOfUse(dvServiceDoc())

## End(Not run)
```

dvUserStudies *Studies in a Dataverse*

Description

Use Data Deposit API to obtain list of studies in a named Dataverse.

Usage

```
dvUserStudies( dvname, dv=getOption('dvn'), user=getOption('dvn.user'),
               pwd=getOption('dvn.pwd'), browser=FALSE, ...)
```

Arguments

dvname	A character string containing the name of a dataverse collection. Must be a dataverse accessible to this user (perhaps as returned by dvServiceDoc.
dv	An optional character string specifying the Dataverse to query. Default is the Harvard IQSS Dataverse.
user	A character string containing a dataverse username. Default is options('dvn.user').
pwd	A character string containing the corresponding dataverse password. Default is options('dvn.pwd').
browser	A logical specifying whether the query should be executed in a web browser. Default is FALSE.
...	Optionally, additional arguments passed to getURL via dvDepositQuery.

Details

For the dvname argument, it is also possible to simply pass the "dvServiceDoc" object directly in lieu of naming a dataverse: dvUserStudies(dvServiceDoc()). If the user has access to more than one dataverse collection, the first will be used (and produce a warning).

Value

A list containing:

dvtitle	The title of the dataverse, as a character string.
released	A character string reporting whether the dataverse is publicly released.
generator	A vector containing details about the SWORD application.
studies	A two-column dataframe containing title and objectId for each study in the dataverse.
xml	A character string containing an XML data structure, which describes the studies available in a named dataverse.

Author(s)

Thomas J. Leeper

See Also

dvStudyStatement dvStudyAtom

Index