

Package ‘dataverse’

June 15, 2017

Version 0.2.0

Date 2017-06-15

Title Client for Dataverse 4 Repositories

Imports stats, utils, httr, xml2, jsonlite

Suggests knitr, testthat, UNF, foreign

Description Provides access to Dataverse version 4 APIs <<https://dataverse.org/>>, enabling data search, retrieval, and deposit. For Dataverse versions <= 4.0, use the deprecated 'dvn' package <<https://cran.r-project.org/package=dvn>>.

License GPL-2

URL <https://github.com/iqss/dataverse-client-r>

BugReports <https://github.com/iqss/dataverse-client-r/issues>

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Thomas J. Leeper [aut, cre],
Jan Kanis [ctb]

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

Repository CRAN

Date/Publication 2017-06-15 15:15:51 UTC

R topics documented:

add_dataset_file	2
add_file	4
create_dataset	5
create_dataverse	7
dataset_atom	8
dataset_versions	9
dataverse	10
dataverse_metadata	11

dataverse_search	12
delete_dataset	14
delete_dataverse	15
delete_file	16
delete_sword_dataset	17
get_dataset	18
get_dataverse	20
get_facets	21
get_file	22
get_user_key	24
initiate_sword_dataset	25
list_datasets	26
publish_dataset	28
publish_dataverse	29
publish_sword_dataset	30
service_document	31
set_dataverse_metadata	32

Index 34

add_dataset_file	<i>Add or update a file in a dataset</i>
------------------	--

Description

Add or update a file in a dataset

Usage

```
add_dataset_file(file, dataset, description = NULL,
  key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

```
update_dataset_file(file, dataset = NULL, id, description = NULL,
  force = TRUE, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

file	A character string
dataset	An integer specifying a dataset identification number or an object of class “dataverse_dataset”. The identification number is the dataset’s persistent identification number (not the integer specifying a specific version of the dataset, such as returned by dataset_versions).
description	Optionally, a character string providing a description of the file.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .

server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .
id	An integer specifying a file identifier; or, if doi is specified, a character string specifying a file name within the DOI-identified dataset; or an object of class "dataverse_file" as returned by dataset_files .
force	A logical indicating whether to force the update even if the file types differ. Default is TRUE.

Details

From Dataverse v4.6.1, the "native" API provides endpoints to add and update files without going through the SWORD workflow. To use SWORD instead, see [add_file](#). `add_dataset_file` adds a new file to a specified dataset.

`update_dataset_file` can be used to replace/update a published file. Note that it only works on published files, so unpublished drafts cannot be updated - the dataset must first either be published ([publish_dataset](#)) or deleted ([delete_dataset](#)).

Value

`add_dataset_file` returns the new file ID.

See Also

[get_dataset](#), [delete_dataset](#), [publish_dataset](#)

Examples

```
## Not run:
meta <- list()
ds <- create_dataset("mydataverse", body = meta)

saveRDS(mtcars, tmp <- tempfile(fileext = ".rds"))
f <- add_dataset_file(tmp, dataset = ds, description = "mtcars")

# publish dataset
publish_dataset(ds)

# update file and republish
saveRDS(iris, tmp)
update_dataset_file(tmp, dataset = ds, id = f,
                    description = "Actually iris")
publish_dataset(ds)

# cleanup
unlink(tmp)
delete_dataset(ds)
```

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

add_file	<i>Add file (SWORD)</i>
----------	-------------------------

Description

Add one or more files to a SWORD (possibly unpublished) dataset

Usage

```
add_file(dataset, file, key = Sys.getenv("DATAVERSE_KEY"),
         server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	A dataset DOI (or other persistent identifier), an object of class “dataset_atom” or “dataset_statement”, or an appropriate and complete SWORD URL.
file	A character vector of file names, a data.frame, or a list of R objects.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function is used to add files to a dataset. It is part of the SWORD API, which is used to upload data to a Dataverse server. This means this can be used to view unpublished Dataverses and Datasets.

As of Dataverse v4.6.1, the “native” API also provides endpoints to add and update files without going through the SWORD workflow. This functionality is provided by [add_dataset_file](#) and [update_dataset_file](#).

Value

An object of class “dataset_atom”.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document
d <- service_document()

# create a list of metadata
metadat <- list(title = "My Study",
                creator = "Doe, John",
                description = "An example study")

# create the dataset
dat <- initiate_sword_dataset("mydataverse", body = metadat)

# add files to dataset
tmp <- tempfile()
write.csv(iris, file = tmp)
f <- add_file(dat, file = tmp)

# publish dataset
publish_dataset(dat)

# delete a dataset
delete_dataset(dat)

## End(Not run)
```

create_dataset	<i>Create or update a dataset</i>
----------------	-----------------------------------

Description

Create or update dataset within a Dataverse

Usage

```
create_dataset(dataverse, body, key = Sys.getenv("DATAVERSE_KEY"),
               server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

```
update_dataset(dataset, body, key = Sys.getenv("DATAVERSE_KEY"),
               server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A character string specifying a Dataverse name or an object of class “dataverse”.
body	A list describing the dataset.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .

server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .
dataset	An integer specifying a dataset identification number or an object of class "dataverse_dataset". The identification number is the dataset's persistent identification number (not the integer specifying a specific version of the dataset, such as returned by dataset_versions).

Details

`create_dataset` creates a Dataverse dataset. In Dataverse, a "dataset" is the lowest-level structure in which to organize files. For example, a Dataverse dataset might contain the files used to reproduce a published article, including data, analysis code, and related materials. Datasets can be organized into "Dataverse" objects, which can be further nested within other Dataverses. For someone creating an archive, this would be the first step to producing said archive (after creating a Dataverse, if one does not already exist). Once files and metadata have been added, the dataset can be published (i.e., made public) using [publish_dataset](#).

`update_dataset` updates a Dataverse dataset that has already been created using [create_dataset](#). This creates a draft version of the dataset or modifies the current draft if one is already in-progress. It does not assign a new version number to the dataset nor does it make it publicly visible (which can be done with [publish_dataset](#)).

Value

An object of class "dataverse_dataset".

See Also

[get_dataset](#), [delete_dataset](#), [publish_dataset](#)

Examples

```
## Not run:
meta <- list()
ds <- create_dataset("mydataverse", body = meta)

meta2 <- list()
update_dataset(ds, body = meta2)

# cleanup
delete_dataset(ds)

## End(Not run)
```

create_dataverse	<i>Create Dataverse</i>
------------------	-------------------------

Description

Create a new Dataverse

Usage

```
create_dataverse(dataverse, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A character string specifying a Dataverse name or an object of class “dataverse”. If missing, a top-level Dataverse is created.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function can create a new Dataverse. In the language of Dataverse, a user has a “root” Dataverse into which they can create further nested Dataverses and/or “datasets” that contain, for example, a set of files for a specific project. Creating a new Dataverse can therefore be a useful way to organize other related Dataverses or sets of related datasets.

For example, if one were involved in an ongoing project that generated monthly data. One may want to store each month’s data and related files in a separate “dataset”, so that each has its own persistent identifier (e.g., DOI), but keep all of these datasets within a named Dataverse so that the project’s files are kept separate the user’s personal Dataverse records. The flexible nesting of Dataverses allows for a number of possible organizational approaches.

Value

A list.

See Also

To manage Dataverses: [delete_dataverse](#), [publish_dataverse](#), [dataverse_contents](#); to get datasets: [get_dataset](#); to search for Dataverses, datasets, or files: [dataverse_search](#)

Examples

```
## Not run:
(dv <- create_dataverse("mydataverse"))

# cleanup
delete_dataverse("mydataverse")

## End(Not run)
```

dataset_atom

View dataset (SWORD)

Description

View a SWORD (possibly unpublished) dataset “statement”

Usage

```
dataset_atom(dataset, key = Sys.getenv("DATAVERSE_KEY"),
             server = Sys.getenv("DATAVERSE_SERVER"), ...)

dataset_statement(dataset, key = Sys.getenv("DATAVERSE_KEY"),
                 server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	A dataset DOI (or other persistent identifier), an object of class “dataset_atom” or “dataset_statement”, or an appropriate and complete SWORD URL.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

These functions are used to view a dataset by its persistent identifier. `dataset_statement` will contain information about the contents of the dataset, whereas `dataset_atom` contains “metadata” relevant to the SWORD API.

Value

A list. For `dataset_atom`, an object of class “dataset_atom”.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_sword_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document
d <- service_document()

# retrieve dataset statement (list contents)
dataset_statement(d[[2]])

# retrieve dataset atom
dataset_atom(d[[2]])

## End(Not run)
```

dataset_versions	<i>Dataset versions</i>
------------------	-------------------------

Description

View versions of a dataset

Usage

```
dataset_versions(dataset, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	An integer specifying a dataset identification number or an object of class “ <code>dataverse_dataset</code> ”. The identification number is the dataset’s persistent identification number (not the integer specifying a specific version of the dataset, such as returned by dataset_versions).
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This returns a list of objects of all versions of a dataset, including metadata. This can be used as a first step for retrieving older versions of files or datasets.

Value

A list of class “dataverse_dataset_version”.

See Also

[get_dataset](#), [dataset_files](#), [publish_dataset](#)

Examples

```
## Not run:
# download file from:
# https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ARKOTI
monogan <- get_dataverse("monogan")
monogan_data <- dataverse_contents(monogan)
d1 <- get_dataset(monogan_data[[1]])
dataset_versions(d1)
dataset_files(d1)

## End(Not run)
```

dataverse

Client for Dataverse 4 Repositories

Description

Provides access to Dataverse 4 APIs, enabling data search, retrieval, and deposit.

Details

Dataverse is open-source data repository management software developed by the Institute for Quantitative Social Science at Harvard University. This package provides an R interface to Dataverse version 4 repositories, including the principal Dataverse hosted at Harvard (<https://dataverse.harvard.edu/>). Users can use the package to search for data stored in a Dataverse repository, retrieve data and other files, and also use the package to directly create and archive their own research data and software.

A Dataverse is structured as a nested set of “dataverse” repositories, such that a single dataverse can contain “datasets” (a set of code files, data files, etc.) or other dataverses. Thus, users may want to search for dataverses (sets of dataverses and datasets), datasets (sets of files), or individual files, and retrieve those objects accordingly. To retrieve a given file, a user typically needs to know what dataset it is stored in. All datasets are identified by a persistent identifier (such as an DOI or Handle, depending on the age of the dataset and what Dataverse repository it is hosted in).

This package provides five main sets of functions to interact with Dataverse:

- Search: [dataverse_search](#)
- Data retrieval: [get_dataverse](#), [dataverse_contents](#), [get_dataset](#), [dataset_metadata](#), [get_file](#)
- Data archiving (SWORD API): [service_document](#), [list_datasets](#), [initiate_sword_dataset](#), [delete_sword_dataset](#), [publish_sword_dataset](#), [add_file](#), [delete_file](#)
- Dataverse management “native” API: [create_dataverse](#), [publish_dataverse](#), [delete_dataverse](#)
- Dataset management “native” API: [create_dataset](#), [update_dataset](#), [publish_dataset](#), [delete_dataset](#), [dataset_files](#), [dataset_versions](#)

References

[Dataverse API Documentation](#)

[Dataverse Homepage](#)

[Harvard IQSS Dataverse](#)

dataverse_metadata *Dataverse metadata*

Description

Get metadata for a named Dataverse.

Usage

```
dataverse_metadata(dataverse, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A character string specifying a Dataverse name or an object of class “dataverse”.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function returns a list of metadata for a named Dataverse. Use [dataverse_contents](#) to list Dataverses and/or datasets contained within a Dataverse or use [dataset_metadata](#) to get metadata for a specific dataset.

Value

A list

See Also

[set_dataverse_metadata](#)

Examples

```
## Not run:
# download file from:
# https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ARKOTI
monogan <- get_dataverse("monogan")
monogan_data <- dataverse_contents(monogan)
dataverse_metadata(monogan)

## End(Not run)
```

dataverse_search	<i>Search Dataverse server</i>
------------------	--------------------------------

Description

Search for Dataverses and datasets

Usage

```
dataverse_search(..., type = c("dataverse", "dataset", "file"),
  subtree = NULL, sort = c("name", "date"), order = c("asc", "desc"),
  per_page = 10, start = NULL, show_relevance = FALSE,
  show_facets = FALSE, fq = NULL, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), verbose = TRUE,
  http_opts = NULL)
```

Arguments

...	A length-one character vector specifying a search query, a named character vector of search arguments, or a sequence of named character arguments. The specific fields available may vary by server installation.
type	A character vector specifying one or more of “dataverse”, “dataset”, and “file”, which is used to restrict the search results. By default, all three types of objects are searched for.
subtree	Currently ignored.
sort	A character vector specifying whether to sort results by “name” or “date”.
order	A character vector specifying either “asc” or “desc” results order.
per_page	An integer specifying the page size of results.

start	An integer specifying used for pagination.
show_relevance	A logical indicating whether or not to show details of which fields were matched by the query
show_facets	A logical indicating whether or not to show facets that can be operated on by the fq parameter
fq	See API documentation.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
verbose	A logical indicating whether to display information about the search query (default is TRUE).
http_opts	Currently ignored.
dataverse	A character string specifying a Dataverse name or an object of class "dataverse".

Details

This function provides an interface for searching for Dataverses, datasets, and/or files within a Dataverse server.

Value

A list.

See Also

[get_file](#), [get_dataverse](#), [get_dataset](#), [dataverse_contents](#)

Examples

```
## Not run:
# simple string search
dataverse_search("Gary King")

# search using named arguments
dataverse_search(c(author = "Gary King", title = "Ecological Inference"))
dataverse_search(author = "Gary King", title = "Ecological Inference")

# search only for datasets
dataverse_search(author = "Gary King", type = "dataset")

## End(Not run)
```

delete_dataset	<i>Delete draft dataset</i>
----------------	-----------------------------

Description

Delete a dataset draft

Usage

```
delete_dataset(dataset, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	An integer specifying a dataset identification number or an object of class “data-verse_dataset”. The identification number is the dataset’s persistent identification number (not the integer specifying a specific version of the dataset, such as returned by dataset_versions).
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function can be used to delete a draft (unpublished) Dataverse dataset. Once published, a dataset cannot be deleted. An existing draft can instead be modified using [update_dataset](#).

Value

A logical.

See Also

[get_dataset](#), [create_dataset](#), [update_dataset](#), [delete_dataset](#), [publish_dataset](#)

Examples

```
## Not run:
meta <- list()
ds <- create_dataset("mydataverse", body = meta)
delete_dataset(ds)

## End(Not run)
```

delete_dataverse	<i>Delete Dataverse</i>
------------------	-------------------------

Description

Delete a dataverse

Usage

```
delete_dataverse(dataverse, key = Sys.getenv("DATAVERSE_KEY"),  
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A character string specifying a Dataverse name or an object of class “dataverse”.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function deletes a Dataverse.

Value

A logical.

See Also

To manage Dataverses: [create_dataverse](#), [publish_dataverse](#), [dataverse_contents](#); to get datasets: [get_dataset](#); to search for Dataverses, datasets, or files: [dataverse_search](#)

Examples

```
## Not run:  
dv <- create_dataverse("mydataverse")  
delete_dataverse(dv)  
  
## End(Not run)
```

delete_file	<i>Delete file (SWORD)</i>
-------------	----------------------------

Description

Delete a file from a SWORD (possibly unpublished) dataset

Usage

```
delete_file(id, key = Sys.getenv("DATAVERSE_KEY"),
            server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

id	A file ID, possibly returned by add_file , or a complete “edit-media/file” URL.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function is used to delete a file from a dataset by its file ID. It is part of the SWORD API, which is used to upload data to a Dataverse server.

Value

If successful, a logical TRUE, else possibly some information.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document
d <- service_document()

# create a list of metadata
metadat <- list(title = "My Study",
```



```

        creator = "Doe, John",
        description = "An example study")

# create the dataset
dat <- initiate_sword_dataset("mydataverse", body = metadat)

# add files to dataset
tmp <- tempfile()
write.csv(iris, file = tmp)
f <- add_file(dat, file = tmp)

# delete a file
ds <- dataset_statement(dat)
delete_file(ds$files[[1]]$id)

# delete a dataset
delete_dataset(dat)

## End(Not run)

```

delete_sword_dataset *Delete dataset (SWORD)*

Description

Delete a SWORD (possibly unpublished) dataset

Usage

```
delete_sword_dataset(dataset, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	A dataset DOI (or other persistent identifier).
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function is used to delete a dataset by its persistent identifier. It is part of the SWORD API, which is used to upload data to a Dataverse server.

Value

If successful, a logical TRUE, else possibly some information.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document
d <- service_document()

# create a list of metadata
metadat <- list(title = "My Study",
               creator = "Doe, John",
               description = "An example study")

# create the dataset in first dataverse
dat <- initiate_sword_dataset(d[[2]], body = metadat)

# delete a dataset
delete_dataset(dat)

## End(Not run)
```

get_dataset

Get dataset

Description

Retrieve a Dataverse dataset or its metadata

Usage

```
get_dataset(dataset, version = ":latest", key = Sys.getenv("DATAVERSE_KEY"),
            server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

```
dataset_metadata(dataset, version = ":latest", block = "citation",
                 key = Sys.getenv("DATAVERSE_KEY"),
                 server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

```
dataset_files(dataset, version = ":latest",
               key = Sys.getenv("DATAVERSE_KEY"),
               server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	An integer specifying a dataset identification number or an object of class “dataverse_dataset”. The identification number is the dataset’s persistent identification number (not the integer specifying a specific version of the dataset, such as returned by dataset_versions).
version	A character string specifying a version of the dataset. This can be one of “:draft” (the current draft), “:latest” (the latest draft, if it exists, or the latest published version), “:latest-published” (the latest published version, ignoring any draft), or “x.y” (where ‘x’ is a major version and ‘y’ is a minor version; the ‘.y’ can be omitted to obtain a major version). In lieu of this, a dataset’s version-specific identification number can be used for the dataset argument.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .
block	A character string specifying a metadata block to retrieve. By default this is “citation”. Other values may be available, depending on the dataset, such as “geospatial” or “socialscience”.

Details

`get_dataset` retrieves details about a Dataverse dataset. `dataset_metadata` returns a named metadata block for a dataset. This is already returned by `get_dataset`, but this function allows you to retrieve just a specific block of metadata, such as citation information. `dataset_files` returns a list of files in a dataset, similar to `get_dataset`. The difference is that this returns only a list of “dataverse_dataset” objects, whereas `get_dataset` returns metadata and a data.frame of files (rather than a list of file objects).

Value

A list of class “dataverse_dataset” or a list of a form dependent on the specific metadata block retrieved. `dataset_files` returns a list of objects of class “dataverse_file”.

See Also

[create_dataset](#), [update_dataset](#), [delete_dataset](#), [publish_dataset](#), [dataset_files](#), [dataset_metadata](#)

Examples

```
## Not run:
# download file from:
# https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ARKOTI
monogan <- get_dataverse("monogan")
monogan_data <- dataverse_contents(monogan)
```

```
d1 <- get_dataset(monogan_data[[1]])
dataset_files(monogan_data[[1]])
f <- get_file(d1$files$datafile$id[3])

## End(Not run)
```

get_dataverse	<i>Get Dataverse</i>
---------------	----------------------

Description

Retrieve details of a Dataverse

Usage

```
get_dataverse(dataverse, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), check = TRUE, ...)

dataverse_contents(dataverse, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A character string specifying a Dataverse name or an object of class “dataverse”.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
check	A logical indicating whether to check that the value of dataverse is actually a numeric
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

`get_dataverse` function retrieves basic information about a Dataverse from a Dataverse server. To see the contents of the Dataverse, use [dataverse_contents](#) instead. Contents might include one or more “datasets” and/or further Dataverses that themselves contain Dataverses and/or datasets. To view the file contents of a single Dataset, use [get_dataset](#).

Value

A list of class “dataverse”.

See Also

To manage Dataverses: [create_dataverse](#), [delete_dataverse](#), [publish_dataverse](#), [dataverse_contents](#);
to get datasets: [get_dataset](#); to search for Dataverses, datasets, or files: [dataverse_search](#)

Examples

```
## Not run:
# view the root dataverse for a server
get_dataverse(":root")
dataverse_contents(":root")

# download file from:
# https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ARKOTI
monogan <- get_dataverse("monogan")
(monogan_data <- dataverse_contents(monogan))

# get a dataset from the dataverse
d1 <- get_dataset(monogan_data[[1]])
f <- get_file(d1$files$datafile$id[3])

## End(Not run)
```

get_facets

Get Dataverse facets

Description

Dataverse metadata facets

Usage

```
get_facets(dataverse, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A character string specifying a Dataverse name or an object of class “dataverse”.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

Retrieve a list of Dataverse metadata facets.

Value

A list.

See Also

To manage Dataverses: [create_dataverse](#), [delete_dataverse](#), [publish_dataverse](#), [dataverse_contents](#); to get datasets: [get_dataset](#); to search for Dataverses, datasets, or files: [dataverse_search](#)

Examples

```
## Not run:
# download file from:
# https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ARKOTI
monogan <- get_dataverse("monogan")
(monogan_data <- dataverse_contents(monogan))

# get facets
get_facets(monogan)

## End(Not run)
```

get_file

Download File(s)

Description

Download Dataverse File(s)

Usage

```
get_file(file, dataset = NULL, format = c("original", "RData", "prep",
    "bundle"), vars = NULL, key = Sys.getenv("DATAVERSE_KEY"),
    server = Sys.getenv("DATAVERSE_SERVER"), ...)

get_file_metadata(file, dataset = NULL, format = c("ddi", "preprocessed"),
    key = Sys.getenv("DATAVERSE_KEY"),
    server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

file An integer specifying a file identifier; or, if doi is specified, a character string specifying a file name within the DOI-identified dataset; or an object of class "dataverse_file" as returned by [dataset_files](#).

dataset	An integer specifying a dataset identification number or an object of class “dataverse_dataset”. The identification number is the dataset’s persistent identification number (not the integer specifying a specific version of the dataset, such as returned by dataset_versions).
format	A character string specifying a file format. For <code>get_file</code> : by default, this is “original” (the original file format). If “RData” or “prep” is used, an alternative is returned. If “bundle”, a compressed directory containing a bundle of file formats is returned. For <code>get_file_metadata</code> , this is “ddi”.
vars	A character vector specifying one or more variable names, used to extract a subset of the data.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function provides access to data files from a Dataverse entry.

Value

`get_file_metadata` returns a character vector containing a DDI metadata file. `get_file` returns a raw vector (or list of raw vectors, if `length(file) > 1`).

Examples

```
## Not run:
# download file from:
# https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ARKOTI
monogan <- get_dataverse("monogan")
monogan_data <- dataverse_contents(monogan)
d1 <- get_dataset("doi:10.7910/DVN/ARKOTI")
f <- get_file(d1$files$datafile$id[3])

# check file metadata
m1 <- get_file_metadata("constructionData.tab", "doi:10.7910/DVN/ARKOTI")
m2 <- get_file_metadata(2437257)

# retrieve file based on DOI and filename
f2 <- get_file("constructionData.tab", "doi:10.7910/DVN/ARKOTI")
f2 <- get_file(2692202)

# retrieve file based on "dataverse_file" object
flist <- dataset_files(2692151)
get_file(flist[[2]])
```

```

# read file as data.frame
if (require("rio")) {
  tmp <- tempfile(fileext = ".dta")
  writeBin(f, tmp)
  str(dat <- rio::import(tmp, haven = FALSE))

  # check UNF match
  #if (require("UNF")) {
  #  unf(dat) %unf% d1$files$datafile$UNF[3]
  #}
}

## End(Not run)

```

get_user_key

Get API Key

Description

Get a user's API key

Usage

```
get_user_key(user, password, server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

user	A character vector specifying a Dataverse server username.
password	A character vector specifying the password for this user.
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

Use a Dataverse server's username and password login to obtain an API key for the user. This can be used if one does not yet have an API key, or desires to reset the key. This function does not require an API key argument to authenticate, but server must still be specified.

Value

A list.

Examples

```
## Not run:
get_user_key("username", "password")

## End(Not run)
```

```
initiate_sword_dataset
      Initiate dataset (SWORD)
```

Description

Initiate a SWORD (possibly unpublished) dataset

Usage

```
initiate_sword_dataset(dataverse, body, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A Dataverse alias or ID number, or an object of class “dataverse”, perhaps as returned by service_document .
body	A list containing one or more metadata fields. Field names must be valid Dublin Core Terms labels (see details, below). The ‘title’, ‘description’, and ‘creator’ fields are required.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function is used to initiate a dataset in a (SWORD) Dataverse by supplying relevant metadata. The function is part of the SWORD API (see [Atom entry specification](#)), which is used to upload data to a Dataverse server. 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”.

Value

An object of class “dataset_atom”.

Note

There are two ways to create dataset: native API ([create_dataset](#)) and SWORD API ([initiate_sword_dataset](#)).

References

[Dublin Core Metadata Terms](#)

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_sword_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document (dataverse list)
d <- service_document()

# create a list of metadata
metadat <- list(title = "My Study",
               creator = "Doe, John",
               description = "An example study")

# create the dataset in first dataverse
dat <- initiate_sword_dataset(d[[2]], body = metadat)

# add files to dataset
tmp <- tempfile(fileext = ".csv")
write.csv(iris, file = tmp)
add_file(dat, file = tmp)

# publish dataset
publish_dataset(dat)

## End(Not run)
```

list_datasets

List datasets (SWORD)

Description

List datasets in a SWORD (possibly unpublished) Dataverse

Usage

```
list_datasets(dataverse, key = Sys.getenv("DATAVERSE_KEY"),
              server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A Dataverse alias or ID number, or an object of class “dataverse”, perhaps as returned by service_document .
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function is used to list datasets in a given Dataverse. It is part of the SWORD API, which is used to upload data to a Dataverse server. This means this can be used to view unpublished Dataverses and Datasets.

Value

A list.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document
d <- service_document()

# list available datasets in first dataverse
list_datasets(d[[2]])

## End(Not run)
```

publish_dataset *Publish dataset*

Description

Publish/release Dataverse dataset

Usage

```
publish_dataset(dataset, minor = TRUE, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	An integer specifying a dataset identification number or an object of class “data-verse_dataset”. The identification number is the dataset’s persistent identification number (not the integer specifying a specific version of the dataset, such as returned by dataset_versions).
minor	A logical specifying whether the new release of the dataset is a “minor” release (TRUE, by default), resulting in a minor version increase (e.g., from 1.1 to 1.2). If FALSE, the dataset is given a “major” release (e.g., from 1.1 to 2.0).
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

Use this function to “publish” (i.e., publicly release) a draft Dataverse dataset. This creates a publicly visible listing of the dataset, accessible by its DOI, with a numbered version. This action cannot be undone. There are no requirements for what constitutes a major or minor release, but a minor release might be used to update metadata (e.g., a new linked publication) or the addition of supplemental files. A major release is best used to reflect a substantial change to the dataset, such as would require a published erratum or a substantial change to data or code.

Value

A list.

See Also

[get_dataset](#), [publish_dataverse](#)

Examples

```
## Not run:
meta <- list()
ds <- create_dataset("mydataverse", body = meta)
publish_dataset(ds)

## End(Not run)
```

publish_dataverse	<i>Publish Dataverse (SWORD)</i>
-------------------	----------------------------------

Description

Publish/re-publish a Dataverse via SWORD

Usage

```
publish_dataverse(dataverse, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	An object of class “sword_collection”, as returned by service_document .
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function is used to publish a (possibly already published) Dataverse. It is part of the SWORD API, which is used to upload data to a Dataverse server.

Value

A list.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

publish_sword_dataset *Publish dataset (SWORD)*

Description

Publish a SWORD (possibly unpublished) dataset

Usage

```
publish_sword_dataset(dataset, key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataset	A dataset DOI (or other persistent identifier), an object of class “dataset_atom” or “dataset_statement”, or an appropriate and complete SWORD URL.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the default is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function is used to publish a dataset by its persistent identifier. This cannot be undone. The function is part of the SWORD API, which is used to upload data to a Dataverse server.

Value

A list.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_sword_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document
d <- service_document()

# create a list of metadata
```

```

metadat <- list(title = "My Study",
               creator = "Doe, John",
               description = "An example study")

# create the dataset in first dataverse
dat <- initiate_sword_dataset(d[[2]], body = metadat)

# publish dataset
publish_sword_dataset(dat)

# delete a dataset
delete_dataset(dat)

## End(Not run)

```

service_document	<i>SWORD Service Document</i>
------------------	-------------------------------

Description

Obtain a SWORD service document.

Usage

```

service_document(key = Sys.getenv("DATAVERSE_KEY"),
                server = Sys.getenv("DATAVERSE_SERVER"), ...)

```

Arguments

key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function can be used to check authentication against the Dataverse SWORD server. It is typically a first step when creating a new Dataverse, a new Dataset, or modifying an existing Dataverse or Dataset.

Value

A list of class "sword_service_document", possibly with one or more "sword_collection" entries. The latter are SWORD representations of a Dataverse. These can be passed to other SWORD API functions, e.g., for creating a new dataset.

See Also

Managing a Dataverse: [publish_dataverse](#); Managing a dataset: [dataset_atom](#), [list_datasets](#), [create_dataset](#), [delete_dataset](#), [publish_dataset](#); Managing files within a dataset: [add_file](#), [delete_file](#)

Examples

```
## Not run:
# retrieve your service document
d <- service_document()

# list available datasets in first dataverse
list_datasets(d[[2]])

## End(Not run)
```

```
set_dataverse_metadata
```

```
Set Dataverse metadata
```

Description

Set Dataverse metadata

Usage

```
set_dataverse_metadata(dataverse, body, root = TRUE,
  key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"), ...)
```

Arguments

dataverse	A character string specifying a Dataverse name or an object of class “dataverse”.
body	A list.
root	A logical.
key	A character string specifying a Dataverse server API key. If one is not specified, functions calling authenticated API endpoints will fail. Keys can be specified atomically or globally using <code>Sys.setenv("DATAVERSE_KEY" = "examplekey")</code> .
server	A character string specifying a Dataverse server. There are multiple Dataverse installations, but the defaults is to use the Harvard Dataverse. This can be modified atomically or globally using <code>Sys.setenv("DATAVERSE_SERVER" = "dataverse.example.com")</code> .
...	Additional arguments passed to an HTTP request function, such as GET , POST , or DELETE .

Details

This function sets the value of metadata fields for a Dataverse. Use [update_dataset](#) to set the metadata fields for a dataset instead.

Value

A list

See Also

[dataverse_metadata](#)

Index

add_dataset_file, [2](#), [4](#)
add_file, [3](#), [4](#), [4](#), [9](#), [11](#), [16](#), [18](#), [26](#), [27](#), [29](#), [30](#),
[32](#)

create_dataset, [4](#), [5](#), [6](#), [9](#), [11](#), [14](#), [16](#), [18](#), [19](#),
[26](#), [27](#), [29](#), [30](#), [32](#)
create_dataverse, [7](#), [11](#), [15](#), [21](#), [22](#)

dataset_atom, [4](#), [8](#), [9](#), [16](#), [18](#), [26](#), [27](#), [29](#), [30](#),
[32](#)
dataset_files, [3](#), [10](#), [11](#), [19](#), [22](#)
dataset_files (get_dataset), [18](#)
dataset_metadata, [11](#), [19](#)
dataset_metadata (get_dataset), [18](#)
dataset_statement (dataset_atom), [8](#)
dataset_versions, [2](#), [6](#), [9](#), [9](#), [11](#), [14](#), [19](#), [23](#),
[28](#)
dataverse, [10](#)
dataverse-package (dataverse), [10](#)
dataverse_contents, [7](#), [11](#), [13](#), [15](#), [20–22](#)
dataverse_contents (get_dataverse), [20](#)
dataverse_metadata, [11](#), [33](#)
dataverse_search, [7](#), [11](#), [12](#), [15](#), [21](#), [22](#)
DELETE, [3](#), [4](#), [6–9](#), [11](#), [14–17](#), [19–21](#), [23–25](#),
[27–32](#)
delete_dataset, [3](#), [4](#), [6](#), [11](#), [14](#), [14](#), [16](#), [19](#),
[27](#), [29](#), [32](#)
delete_dataverse, [7](#), [11](#), [15](#), [21](#), [22](#)
delete_file, [4](#), [9](#), [11](#), [16](#), [16](#), [18](#), [26](#), [27](#), [29](#),
[30](#), [32](#)
delete_sword_dataset, [9](#), [11](#), [17](#), [26](#), [30](#)

GET, [3](#), [4](#), [6–9](#), [11](#), [14–17](#), [19–21](#), [23–25](#), [27–32](#)
get_dataset, [3](#), [6](#), [7](#), [10](#), [11](#), [13–15](#), [18](#),
[19–22](#), [28](#)
get_dataverse, [11](#), [13](#), [20](#)
get_facets, [21](#)
get_file, [11](#), [13](#), [22](#)
get_file_metadata (get_file), [22](#)
get_user_key, [24](#)

initiate_sword_dataset, [11](#), [25](#)

list_datasets, [4](#), [9](#), [11](#), [16](#), [18](#), [26](#), [26](#), [27](#),
[29](#), [30](#), [32](#)

POST, [3](#), [4](#), [6–9](#), [11](#), [14–17](#), [19–21](#), [23–25](#),
[27–32](#)
publish_dataset, [3](#), [4](#), [6](#), [9–11](#), [14](#), [16](#), [18](#),
[19](#), [26](#), [27](#), [28](#), [29](#), [30](#), [32](#)
publish_dataverse, [4](#), [7](#), [9](#), [11](#), [15](#), [16](#), [18](#),
[21](#), [22](#), [26–29](#), [29](#), [30](#), [32](#)
publish_sword_dataset, [11](#), [30](#)

service_document, [11](#), [25](#), [27](#), [29](#), [31](#)
set_dataverse_metadata, [12](#), [32](#)

update_dataset, [11](#), [14](#), [19](#), [33](#)
update_dataset (create_dataset), [5](#)
update_dataset_file, [4](#)
update_dataset_file (add_dataset_file),
[2](#)