Package 'dataverse'

March 23, 2023

Version 0.3.13

Title Client for Dataverse 4+ Repositories

Imports checkmate, httr, jsonlite, readr, stats, utils, xml2

Suggests covr, haven, knitr, purrr, rmarkdown, testthat, devtools, tibble, yaml

Description Provides access to Dataverse APIs <https://dataverse.org/> (versions 4-5), enabling data search, retrieval, and deposit. For Dataverse versions <= 3.0, use the archived 'dvn' package <https://cran.r-project.org/package=dvn>.

License GPL-2

URL https://iqss.github.io/dataverse-client-r/, https://dataverse.org/, https://github.com/iqss/dataverse-client-r

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

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.0

Config/testthat/edition 3

NeedsCompilation no

Author Shiro Kuriwaki [aut, cre] (<https://orcid.org/0000-0002-5687-2647>), Will Beasley [aut] (<https://orcid.org/0000-0002-5613-5006>), Thomas J. Leeper [aut] (<https://orcid.org/0000-0003-4097-6326>), Philip Durbin [aut] (<https://orcid.org/0000-0002-9528-9470>), Sebastian Karcher [aut] (<https://orcid.org/0000-0001-8249-7388>), Jan Kanis [ctb], Edward Jee [ctb]

Maintainer Shiro Kuriwaki <shirokuriwaki@gmail.com>

Repository CRAN

Date/Publication 2023-03-23 15:42:05 UTC

R topics documented:

`add_dataset_file` *Add or update a file in a dataset*

Description

Add or update a file in a dataset. For most applications, this is the recommended function to upload your own local datasets to an existing Dataverse dataset. Uploading requires a Dataverse API Key in the `key` variable.

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 for the location path of the file to be uploaded.
dataset	A character specifying a persistent identification ID for a dataset, for example "doi:10.70122/FK2/HXJVJU". Alternatively, an object of class "dataverse_dataset" obtained by dataverse_contents().
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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
...	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. It also uploads the file to the dataset.

See Also

`get_dataset`, `delete_dataset`, `publish_dataset`

Examples

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


# Upload RDS dataset saved to local
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` *Add file (SWORD)*

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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
...	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 *Create or update a dataset*

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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
...	Additional arguments passed to an HTTP request function, such as `GET`, `POST`, or `DELETE`.
dataset	A character specifying a persistent identification ID for a dataset, for example `"doi:10.70122/FK2/HXJVJU"`. Alternatively, an object of class "dataverse_dataset" obtained by `dataverse_contents()`.

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` *Create Dataverse*

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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`...`	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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
...	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 *Dataset versions*

Description

View versions of a dataset

Usage

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

Arguments

dataset	A character specifying a persistent identification ID for a dataset, for example "doi:10.70122/FK2/HXJVJU". Alternatively, an object of class "dataverse_dataset" obtained by dataverse_contents().
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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
...	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 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 download: `get_dataframe_by_name`, `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

Documentation for this R Package

Code Repository for the R Package

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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`...`	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 *Search Dataverse server*

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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
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 *Delete draft dataset*

Description

Delete a dataset draft

Usage

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

Arguments

dataset	A character specifying a persistent identification ID for a dataset, for example "doi:10.70122/FK2/HXJVJU". Alternatively, an object of class "dataverse_dataset" obtained by dataverse_contents().
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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
...	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 *Delete Dataverse*

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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" =

"dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.

... 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 *Delete file (SWORD)*

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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
...	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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`...`	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_dataframe_by_name *Download dataverse file as a dataframe*

Description

Reads in the Dataverse file into the R environment with any user-specified function, such as read.csv or readr functions.

Use get_dataframe_by_name if you know the name of the datafile and the DOI of the dataset. Use get_dataframe_by_doi if you know the DOI of the datafile itself. Use get_dataframe_by_id if you know the numeric ID of the datafile. For files that are not datasets, the more generic get_file that downloads the content as a binary is simpler.

The function can read datasets that are unpublished and are still drafts, as long as the entry has a UNF. See the download vignette for details.

Usage

```
get_dataframe_by_name(
  filename,
  dataset = NULL,
  .f = NULL,
```

```
  original = FALSE,
  ...
)

get_dataframe_by_id(fileid, .f = NULL, original = FALSE, ...)

get_dataframe_by_doi(filedoi, .f = NULL, original = FALSE, ...)
```

Arguments

filename	The name of the file of interest, with file extension, for example "roster-bulls-1996.tab". Can be a vector for multiple files.
dataset	A character specifying a persistent identification ID for a dataset, for example "doi:10.70122/FK2/HXJVJU". Alternatively, an object of class "dataverse_dataset" obtained by dataverse_contents().
.f	The function to used for reading in the raw dataset. The user must choose the appropriate function: for example if the target is a .rds file, then .f should be readRDS or readr::read_rds. It can be a custom function defined by the user. See examples for details.
original	A logical, whether to read the ingested, archival version of the datafile if one exists. If TRUE, users should supply a function to use to read in the original. The archival versions are tab-delimited .tab files so if original = FALSE, .f is set to readr::read_tsv.
...	Arguments passed on to get_file

file An integer specifying a file identifier; or a vector of integers specifying file identifiers; or, if used with the prefix "doi:", a character with the file-specific DOI; or, if used without the prefix, a filename accompanied by a dataset DOI in the dataset argument, or an object of class "dataverse_file" as returned by dataset_files. Can be a vector for multiple files.

format A character string specifying a file format for download. by default, this is "original" (the original file format). If NULL, no query is added, so ingested files are returned in their ingested TSV form. For tabular datasets, the option "bundle" downloads the bundle of the original and archival versions, as well as the documentation. See https://guides.dataverse.org/en/latest/api/dataaccess.html for details.

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 Sys.setenv("DATAVERSE_KEY" = "examplekey").

server A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu

in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.

fileid — A numeric ID internally used for get_file_by_id. Can be a vector for multiple files.

filedoi — A DOI for a single file (not the entire dataset), of the form "10.70122/FK2/PPIAXE/MHDB0O" or "doi:10.70122/FK2/PPIAXE/MHDB0O". Can be a vector for multiple files.

Value

A R object that is returned by the default or user-supplied function .f argument. For example, if .f = readr::read_tsv(), the function will return a dataframe as read in by readr::read_tsv(). If the file identifier is a vector, it will return a list where each slot corresponds to elements of the vector.

Examples

```
## Not run:
# 1. For files originally in plain-text (.csv, .tsv), we recommend
# retreiving data.frame from dataverse DOI and file name, or the file's DOI.

df_tab <-
  get_dataframe_by_name(
    filename = "roster-bulls-1996.tab",
    dataset  = "doi:10.70122/FK2/HXJVJU",
    server   = "demo.dataverse.org"
  )

df_tab <-
  get_dataframe_by_doi(
    filedoi      = "10.70122/FK2/HXJVJU/SA3Z2V",
    server       = "demo.dataverse.org"
  )

# 2. For files where Dataverse's ingest loses information (Stata .dta, SPSS .sav)
# or cannot be ingested (R .rds), we recommend
# specifying `original = TRUE` and specifying a read-in function in .f.

# Rds files are not ingested so original = TRUE and .f is required.
if (requireNamespace("readr", quietly = TRUE)) {
  df_from_rds_original <-
    get_dataframe_by_name(
      filename   = "nlsw88_rds-export.rds",
      dataset    = "doi:10.70122/FK2/PPIAXE",
      server     = "demo.dataverse.org",
      original   = TRUE,
      .f         = readr::read_rds
    )
}

# Stata dta files lose attributes such as value labels upon ingest so
# reading the original version by a Stata reader such as `haven` is recommended.
```

```
if (requireNamespace("haven", quietly = TRUE)) {
  df_stata_original <-
    get_dataframe_by_name(
      filename   = "nlsw88.tab",
      dataset    = "doi:10.70122/FK2/PPIAXE",
      server     = "demo.dataverse.org",
      original   = TRUE,
      .f         = haven::read_dta
    )
}

# 3. RData files are read in by `base::load()` but cannot be assigned to an
# object name. The following shows two possible ways to read in such files.

# First, without relying on `get_dataframe_*`, write as a binary file:
as_binary <- get_file_by_doi(
 filedoi = "doi:10.70122/FK2/PPIAXE/5VPXKE",
 server = "demo.dataverse.org")

temp <- tempdir()
writeBin(as_binary, path(temp, "county.RData"))
load(path(temp, "county.RData"))

# If you are certain each RData contains only one object, one could define a
# custom function used in https://stackoverflow.com/a/34926943
load_object <- function(file) {
  tmp <- new.env()
  load(file = file, envir = tmp)
  tmp[[ls(tmp)[1]]]
}

# https://demo.dataverse.org/file.xhtml?persistentId=doi:10.70122/FK2/PPIAXE/X2FC5V
as_rda <- get_dataframe_by_id(
  file = 1939003,
  server = "demo.dataverse.org",
  .f = load_object,
  original = TRUE)

## End(Not run)
```

`get_dataset` *Get dataset metadata*

Description

Retrieve metadata. To actually download a data file, see `get_file` or `get_dataframe_by_name`.

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	A character specifying a persistent identification ID for a dataset, for example "doi:10.70122/FK2/HXJVJU". Alternatively, an object of class "dataverse_dataset" obtained by dataverse_contents().
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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.

...	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

get_file

Examples

```
## Not run:
# https://demo.dataverse.org/dataverse/dataverse-client-r
Sys.setenv("DATAVERSE_SERVER" = "demo.dataverse.org")

# download file from:
dv <- get_dataverse("dataverse-client-r")
contents <- dataverse_contents(dv)[[1]]

dataset_files(contents[[1]])
get_dataset(contents[[1]])
dataset_metadata(contents[[1]])

Sys.unsetenv("DATAVERSE_SERVER")

## End(Not run)
```

`get_dataverse` *Get Dataverse*

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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`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".

Examples

```
## Not run:
# https://demo.dataverse.org/dataverse/dataverse-client-r
Sys.setenv("DATAVERSE_SERVER" = "demo.dataverse.org")

# download file from:
dv <- get_dataverse("dataverse-client-r")

# get a dataset from the dataverse
(d1 <- get_dataset(dataverse_contents(dv)[[1]]))

# download a file using the metadata
get_dataframe_by_name("roster-bulls-1996.tab", d1$datasetPersistentId)

## 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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.

server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
...	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 Dataverse file as a raw binary*

Description

Download Dataverse File(s). get_file_* functions return a raw binary file, which cannot be readily analyzed in R. To use the objects as dataframes, see the get_dataframe_* functions at ?get_dataframe instead.

Usage

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

get_file_by_name(
  filename,
  dataset,
  format = c("original", "bundle"),
  vars = NULL,
  key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"),
  original = TRUE,
  ...
)

get_file_by_id(
  fileid,
  dataset = NULL,
  format = c("original", "bundle"),
  vars = NULL,
  original = TRUE,
  progress = NULL,
  key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"),
  ...
)

get_file_by_doi(
  filedoi,
  dataset = NULL,
  format = c("original", "bundle"),
  vars = NULL,
  original = TRUE,
  key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"),
  ...
)
```

Arguments

file	An integer specifying a file identifier; or a vector of integers specifying file identifiers; or, if used with the prefix "doi:", a character with the file-specific DOI; or, if used without the prefix, a filename accompanied by a dataset DOI in the dataset argument, or an object of class "dataverse_file" as returned by dataset_files. Can be a vector for multiple files.
dataset	A character specifying a persistent identification ID for a dataset, for example "doi:10.70122/FK2/HXJVJU". Alternatively, an object of class "dataverse_dataset" obtained by dataverse_contents().
format	A character string specifying a file format for download. by default, this is "original" (the original file format). If NULL, no query is added, so ingested files are returned in their ingested TSV form. For tabular datasets, the option "bundle" downloads the bundle of the original and archival versions, as well as the documentation. See https://guides.dataverse.org/en/latest/api/dataaccess.html for details.
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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
original	A logical, defaulting to TRUE. If a ingested (.tab) version is available, download the original version instead of the ingested? If there was no ingested version, is set to NA. Note in get_dataframe_*, original is set to FALSE by default. Either can be changed.
...	Additional arguments passed to an HTTP request function, such as GET, POST, or DELETE.
filename	Filename of the dataset, with file extension as shown in Dataverse (for example, if nlsw88.dta was the original but is displayed as the ingested nlsw88.tab, use the ingested version.)
fileid	A numeric ID internally used for get_file_by_id. Can be a vector for multiple files.
progress	Whether to show a progress bar of the download. If not specified, will be set to TRUE for a file larger than 100MB. To fix a value, set FALSE or TRUE.
filedoi	A DOI for a single file (not the entire dataset), of the form "10.70122/FK2/PPIAXE/MHDB0O" or "doi:10.70122/FK2/PPIAXE/MHDB0O". Can be a vector for multiple files.

Details

This function provides access to data files from a Dataverse entry. `get_file` is a general wrapper, and can take either dataverse objects, file IDs, or a filename and dataverse. Internally, all functions download each file by `get_file_by_id`. `get_file_by_name` is a shorthand for running `get_file` by specifying a file name (`filename`) and dataset (`dataset`). `get_file_by_doi` obtains a file by its file DOI, bypassing the `dataset` argument.

Value

`get_file` returns a raw vector (or list of raw vectors, if `length(file) > 1`), which can be saved locally with the `writeBin` function. To load datasets into the R environment dataframe, see get_dataframe_by_name.

See Also

To load the objects as datasets get_dataframe_by_name.

Examples

```
## Not run:

# 1. Using filename and dataverse
f1 <- get_file_by_name(
  filename = "nlsw88.tab",
  dataset  = "10.70122/FK2/PPIAXE",
  server   = "demo.dataverse.org"
)

# 2. Using file DOI
f2 <- get_file_by_doi(
  filedoi  = "10.70122/FK2/PPIAXE/MHDB0O",
  server   = "demo.dataverse.org"
)

# 3. Two-steps: Find ID from get_dataset
d3 <- get_dataset("doi:10.70122/FK2/PPIAXE", server = "demo.dataverse.org")
f3 <- get_file(d3$files$id[1], server = "demo.dataverse.org")

# 4. Retrieve multiple raw data in list
f4_meta <- get_dataset(
  "doi:10.70122/FK2/PPIAXE",
  server = "demo.dataverse.org"
)

f4 <- get_file(f4_meta$files$id, server = "demo.dataverse.org")
names(f4) <- f4_meta$files$label

# Write binary files. To load into R environment, use get_dataframe_by_name()
# The appropriate file extension needs to be assigned by the user.

writeBin(f1, "nlsw88.dta") # .tab extension but save as dta
writeBin(f4[["nlsw88_rds-export.rds"]], "nlsw88.rds") # originally a rds file
```

```
writeBin(f4[["nlsw88.tab"]], "nlsw88.dta") # originally a dta file

## End(Not run)
```

`get_file_metadata` *Retrieve a ddi metadata file*

Description

Retrieve a ddi metadata file

Usage

```
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 a vector of integers specifying file identifiers; or, if used with the prefix `"doi:"`, a character with the file-specific DOI; or, if used without the prefix, a filename accompanied by a dataset DOI in the `dataset` argument, or an object of class "dataverse_file" as returned by `dataset_files`. Can be a vector for multiple files.
`dataset`	A character specifying a persistent identification ID for a dataset, for example `"doi:10.70122/FK2/HXJVJU"`. Alternatively, an object of class "dataverse_dataset" obtained by `dataverse_contents()`.
`format`	Defaults to "ddi" for metadata files
`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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`...`	Additional arguments passed to an HTTP request function, such as `GET`, `POST`, or `DELETE`.

Value

A character vector containing a DDI metadata file.

Examples

```
## Not run:
 ddi_raw <- get_file_metadata(file = "nlsw88.tab",
                              dataset = "10.70122/FK2/PPIAXE",
                              server = "demo.dataverse.org")
 xml2::read_xml(ddi_raw)

## 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	The Dataverse instance. See get_file.
...	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:
 # Replace Username and password with personal login
 get_user_key("username", "password", server = "dataverse.harvard.edu")

## 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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
...	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)
```

`is_ingested` *Identify if file is an ingested file*

Description

Identify if file is an ingested file

Usage

```
is_ingested(
  x,
  key = Sys.getenv("DATAVERSE_KEY"),
  server = Sys.getenv("DATAVERSE_SERVER"),
  ...
)
```

Arguments

`x`	A numeric fileid or file-specific DOI
`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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`...`	Arguments passed on to `get_file` (no effect here)

Value

Length-1 logical, `TRUE` if it is ingested and `FALSE` otherwise

Examples

```
## Not run:
# https://demo.dataverse.org/file.xhtml?persistentId=doi:10.70122/FK2/PPIAXE
# nlsw88.tab
is_ingested(x = "doi:10.70122/FK2/PPIAXE/MHDB0O",
            server = "demo.dataverse.org")
```

```
is_ingested(x = 1734017,
            server = "demo.dataverse.org")

# nlsw88_rds-export.rds
is_ingested(x = "doi:10.70122/FK2/PPIAXE/SUCFNI",
            server = "demo.dataverse.org")
is_ingested(x = 1734016,
            server = "demo.dataverse.org")

## 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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`...`	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:
Sys.setenv("DATAVERSE_SERVER" = "demo.dataverse.org")
Sys.setenv("DATAVERSE_KEY"    = "c7208dd2-6ec5-469a-bec5-f57e164888d4")
dv <- get_dataverse("dataverse-client-r")
list_datasets(dv)

## 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	A character specifying a persistent identification ID for a dataset, for example "doi:10.70122/FK2/HXJVJU". Alternatively, an object of class "dataverse_dataset" obtained by dataverse_contents().
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 Sys.setenv("DATAVERSE_KEY" = "examplekey").

server — A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.

... — 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 *Publish Dataverse (SWORD)*

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 Sys.setenv("DATAVERSE_KEY" = "examplekey").
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with "dataverse.harvard.edu" being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu") or add DATAVERSE_SERVER = "dataverse.harvard.edu in one's .Renviron file (usethis::edit_r_environ()), with the appropriate domain as its value.
...	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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
server	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
...	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` *SWORD Service Document*

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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.
`server`	A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.
`...`	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 `Sys.setenv("DATAVERSE_KEY" = "examplekey")`.

server — A character string specifying a Dataverse server. Multiple Dataverse installations exist, with `"dataverse.harvard.edu"` being the most major. The server can be defined each time within a function, or it can be set as a default via an environment variable. To set a default, run `Sys.setenv("DATAVERSE_SERVER" = "dataverse.harvard.edu")` or add `DATAVERSE_SERVER = "dataverse.harvard.edu` in one's `.Renviron` file (`usethis::edit_r_environ()`), with the appropriate domain as its value.

... — 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