Package ‘rfigshare’

May 9, 2022

Title An R Interface to ‘figshare’

Description An R interface to ‘figshare’.

Version 0.3.8

License CC0

URL https://github.com/ropensci-archive/rfigshare

BugReports https://github.com/ropensci-archive/rfigshare/issues

Suggests testthat, gdata, uuid, knitr, rmarkdown

Depends R (>= 3.0), methods

Imports RJSONIO, httr (>= 0.3), httpuv, yaml, ggplot2, XML

VignetteBuilder knitr

RoxygenNote 7.1.2

X-schema.org-applicationCategory Data Publication

X-schema.org-keywords data, archiving, repository, doi

X-schema.org-isPartOf https://ropensci.org

NeedsCompilation no

Author Carl Boettiger [aut, cre] (<https://orcid.org/0000-0002-1642-628X>),
Scott Chamberlain [aut] (<https://orcid.org/0000-0003-1444-9135>),
Karthik Ram [aut] (<https://orcid.org/0000-0002-0233-1757>),
Edmund Hart [aut] (<https://orcid.org/0000-0001-7367-7969>)

Maintainer Carl Boettiger <cboettig@gmail.com>

Repository CRAN

Date/Publication 2022-05-09 19:40:02 UTC

R topics documented:

  fs_add_authors ........................................... 2
  fs_add_categories ......................................... 3
  fs_add_links ............................................. 4
  fs_auth ..................................................... 5
fs_add_authors

Description
Add author to an article

Usage
fs_add_authors(
    article_id,
    authors,
    session = fs_get_auth(),
    create_missing = TRUE,
    debug = FALSE
)

Arguments
article_id id number of an article on figshare
authors a list or character string of authors or author id numbers (or mixed).
session (optional) the authentication credentials from fs_auth. If not provided, will attempt to load from cache as long as figshare_auth has been run.
create_missing (logical) Attempt to create authors not already registered on FigShare?
debug return the httr result visibly?
Value

adds the requested authors to the given article

Author(s)

Carl Boettiger < cboettig@gmail.com >

Examples

```r
## Not run:
fs_add_authors("138", list("Scott Chamberlain", "Karthik Ram"))
fs_add_authors("138", c("Scott Chamberlain", "Karthik Ram"))
fs_add_authors("138", list("Scott Chamberlain", "97306"))
fs_add_authors("138", list("Scott Chamberlain", 97306))
fs_add_authors(138, 97306)

## End(Not run)
```

---

```r
fs_add_categories  Add a category to article
```

Description

Add a category to article

Usage

```r
fs_add_categories(
  article_id,
  category_id,
  session = fs_get_auth(),
  debug = FALSE
)
```

Arguments

- `article_id` the id number of the article
- `category_id` is a vector of integers corresponding to categories or a vector of category names
- `session` (optional) the authentication credentials from `fs_auth`. If not provided, will attempt to load from cache as long as `figshare_auth` has been run.
- `debug` return PUT results visibly?

Value

output of PUT request (invisibly)
### fs_add_links

**Add link to article**

**Description**

Adds url links to the metadata of an article

**Usage**

```r
fs_add_links(article_id, link, session = fs_get_auth(), debug = FALSE)
```

**Arguments**

- `article_id` the id number of the article
- `link` the url you wish to add (can be list of urls)
- `session` (optional) the authentication credentials from `fs_auth`. If not provided, will attempt to load from cache as long as authentication has been run.
- `debug` logical, should function return details of PUT request?

**Value**

output of PUT request (invisibly)

**Author(s)**

Carl Boettiger <cboettig@gmail.com>

**References**

https://docs.figshare.com
See Also

fs_auth

Examples

## Not run:
fs_add_links(138, list("http://carlboettiger.info", "http://ropensci.org"))

## End(Not run)

### fs_auth

Figshare authentication via OAuth1.0 using httr

Description

Figshare authentication via OAuth1.0 using httr

Usage

fs_auth(
  cKey = getOption("FigshareKey", NULL),
  cSecret = getOption("FigsharePrivateKey", NULL),
  token = getOption("FigshareToken", NULL),
  token_secret = getOption("FigsharePrivateToken", NULL)
)

Arguments

cKey optional argument for the consumer key. See details.
cSecret optional argument for the consumer secret key. See details.
token optional argument for the user-specific token. See details.
token_secret Optional argument to provide a secret token assigned to the user, rather than let fs_auth() automatically handle authentication. See details.

Details

Explicit calls to fs_auth() are usually not needed, as the function is called automatically by all other functions that need authentication. As of version 0.3, no arguments are needed as authentication is done online, and fs_auth() will not attempt to load keys stored in options.

By default, the function will use the application’s consumer key and consumer secret key, rather than expecting the user to create their own application. The user-specific tokens will then be generated and locally cached for use between sessions, if indicated by the interactive options. For details, see httr oauth1.0_token documentation.

If for some reason a user would rather provide there token and secret token as before this is still supported using the same arguments. Users wanting to have their own app can provide cKey and cSecret arguments too, but this is provided primarily for backwards compatibility with older versions. It is expected that most users will leave the keys as NULL.
Value

OAuth credential (invisibly). The credential is also written to the environment "FigshareAuthCache", which is created when the package is loaded. All functions needing authentication can then access the credential without it being explicitly passed in the function call. If authentication fails, returns the failing GET response for debugging.

Author(s)

Carl Boettiger <cboettig@gmail.com>

References

https://docs.figshare.com

Examples

## Not run:
fs_auth()
## End(Not run)

---

**fs_author_search**  
*Search for an author*

---

**Description**

Function to search for authors

**Usage**

fs_author_search(author, session = fs_get_auth(), debug = FALSE)

**Arguments**

- **author**: a string to search for (name, can include spaces)
- **session**: (optional) the authentication credentials from `fs_auth`. If not provided, will attempt to load from cache as long as figshare_auth has been run.
- **debug**: toggle debugging mode

**Value**

output of PUT request (invisibly)

**Author(s)**

Carl Boettiger <cboettig@gmail.com>
Browse can be set to all public articles, the users own articles. Browse can filter on matching timestamp, author, title, description, tag, category, and date range.

Usage

```r
fs_browse(
  mine = TRUE,
  public_only = FALSE,
  private_only = FALSE,
  drafts_only = FALSE,
  session = fs_get_auth()
)
```

**Arguments**

- **mine** Logical, show only my (authenticated user’s) articles. Defaults to TRUE.
- **public_only** (for use with mine=TRUE only) browse only my public articles. default is FALSE
- **private_only** (for use with mine=TRUE only) browse only my private articles. default is FALSE
- **drafts_only** (for use with mine=TRUE only) browse only my draft articles. default is FALSE
- **session** (optional) the authentication credentials from `fs_auth`. If not provided, will attempt to load from cache as long as figshare_auth has been run.
- **base** the API access url
- **query** a search query term (equivalent to calling `fs_search`)
- **debug** enable debugging mode
Value

output of PUT request (invisibly)

Author(s)

Carl Boettiger <cboettig@gmail.com>

References

https://docs.figshare.com

See Also

fs_auth

Examples

## Not run:
fs_browse()

## End(Not run)

---

fs_category_list List all categories

Description

List all categories

Usage

fs_category_list(debug = FALSE)

Arguments

d debug enable debugging

Value

a table of all the categories

References

https://docs.figshare.com
Examples

```r
## Not run:
fs_categories_list()

## End(Not run)
```

---

**fs_cat_to_id**

*Helper function that matches string categories to id’s*

**Description**

Helper function that matches string categories to id’s

**Usage**

```r
fs_cat_to_id(category_id)
```

**Arguments**

- `category_id`
  - Must be a valid category string, regardless of case

**Value**

- a vector of integers corresponding to valid figshare categories

**Author(s)**

Edmund Hart <edmund.m.hart@gmail.com>

**References**

https://docs.figshare.com

---

**fs_create**

*Create a FigShare article (draft)*

**Description**

Articles must be created with `fs_create` with essential metadata. Then you can add files with `fs_upload`, add categories, tags or authors with `fs_add_categories` or `fs_add_tags` `fs_add_authors`. Authors not registered with a FigShare id can be created with `fs_create_author`. You can edit the original metadata with `fs_update`. Finally, release the article as either private or public with `fs_make_private` or `fs_make_public`. Before creating the article, you must authenticate using `fs_auth`. 
Usage

```r
fs_create(
  title,
  description,
  type = c("dataset", "figure", "media", "poster", "paper", "fileset"),
  session = fs_get_auth(),
  debug = FALSE
)
```

Arguments

- **title**: for the article
- **description**: of the article
- **type**: one of: dataset, figure, media, poster, paper or fileset. (Only filesets can have multiple uploaded files attached).
- **session**: the authentication credentials from `fs_auth`
- **debug**: print full post call return

Value

article id

References

[https://docs.figshare.com](https://docs.figshare.com)

See Also

`fs_auth`, `fs_upload`

Examples

```r
## Not run:
fs_create("My Title", "A description of the object", "dataset")
## End(Not run)
```

---

**fs_create_author**

*Creates a figshare author*

### Description

Creates a figshare author

### Usage

```r
fs_create_author(full_name, session = fs_get_auth(), debug = FALSE)
```
**fs_delete**

Delete article (private or drafts only) or attached file

### Description

Delete article (private or drafts only) or attached file

#### Usage

```r
fs_delete(article_id, file_id = NULL, session = fs_get_auth(), debug = FALSE)
```

#### Arguments

- `article_id`: the id number of the article
- `file_id`: the id number of the file, if removing an attached file from a fileset. `file_id` defaults to NULL, removing the entire draft or private article.
- `session`: (optional) the authentication credentials from `fs_auth`. If not provided, will attempt to load from cache as long as figshare_auth has been run.
- `debug`: display return value of request?
Value

output of DELETE request (invisibly)

Author(s)

Carl Boettiger <cboettig@gmail.com>

References

https://docs.figshare.com

See Also

fs_auth

Examples

```r
## Not run:
fs_delete(123)

## Delete all attachments in the second-most-recent entry in my library
my_lib <- fs_browse(mine=TRUE)
article_id <- my_lib[[2]]$article_id
file_ids <- sapply(my_lib[[2]]$files, `[[`, "id")
sapply(file_ids, function(id) fs_delete(article_id, id))

## End(Not run)
```

### fs_details

**Get details for an article**

**Description**

Get details for an article

**Usage**

```r
fs_details(
  article_id,
  mine = is_mine(article_id),
  session = fs_get_auth(),
  show_versions = FALSE,
  version = NULL,
  debug = FALSE
)
```
## fs_download

### Get details for an article

**Description**

Get details for an article

**Usage**

```r
fs_download(
  article_id,
  urls_only = TRUE,
  mine = is_mine(article_id),
  session = fs_get_auth(),
  show_versions = FALSE,
  version = NULL,
  ...  
)
```

### Arguments

- `article_id` number
- `mine` logical (default FALSE). Set to true to see article details for your own non-public articles
- `session` the authentication credentials from `fs_auth`
- `show_versions` logical, show what versions are available
- `version` show a given version number
- `debug` logical, enable debugging mode?

### Author(s)

Carl Boettiger <cboettig@gmail.com>

### References

[https://docs.figshare.com](https://docs.figshare.com)

### See Also

- `fs_auth`

### Examples

```r
## Not run:
fs_details(138)

## End(Not run)
```
Arguments

- **article_id**: number
- **urls_only**: logical (default TRUE) to only return the URLs to the downloadable objects but do not call download.file. If FALSE, will download files
- **mine**: logical (default FALSE). Set to true to see article details for your own non-public articles
- **session**: the authentication credentials from `fs_auth`
- **show_versions**: logical, show what versions are available
- **version**: show a given version number
- **...**: additional arguments to `download.file`

Author(s)

Carl Boettiger <cboettig@gmail.com>

References

- https://docs.figshare.com
- https://github.com/ropensci-archive/rfigshare

See Also

- `fs_auth`
- `download.file`

Examples

```r
## Not run:
url <- fs_download(90818)
data <- read.csv(url)
articles <- fs_search("SciFund")
ids <- fs_ids(articles)
fs_download(ids, urls_only=FALSE)
## End(Not run)
```

Description

Upload a figure to figshare and return the url

Usage

```r
fs_embed(file)
```
fs_ids

Arguments

file path to an image file

Details

use with opts_knit$set(upload.fn = fs_embed)

Value

a url to the image file

Description

Get a list of article id numbers from a search return

Usage

fs_ids(object)

Arguments

object the output of a search

Value

a list of article id numbers

Author(s)

Carl Boettiger <cboettig@gmail.com>

References

https://docs.figshare.com

Examples

## Not run:
figshare_category()

## End(Not run)
### fs_image_url

**get image url**

**Description**

get image url

**Usage**

```r
fs_image_url(id, debug = FALSE)
```

**Arguments**

- `id`: a (public) figshare figure id number
- `debug`: logical, enable debugging mode?

**Details**

this is currently an unstable hack of html parsing

**Value**

a url to the image file

---

### fs_make_private

**Make an article private (draft only?)**

**Description**

Make an article private (draft only?)

**Usage**

```r
fs_make_private(article_id, session = fs_get_auth())
```

**Arguments**

- `article_id`: the id number of the article
- `session`: (optional) the authentication credentials from `fs_auth`. If not provided, will attempt to load from cache as long as `fs_auth` has been run.

**Value**

output of PUT request (invisibly)
Author(s)
Carl Boettiger <cboettig@gmail.com>

References
https://docs.figshare.com

See Also
fs_auth

Examples

```r
## Not run:
fs_make_private(123)

## End(Not run)
```

---

### fs_make_public

Make an article public (for private or draft articles)

#### Description
Make an article public (for private or draft articles)

#### Usage

```r
fs_make_public(article_id, session = fs_get_auth())
```

#### Arguments

- `article_id`: the id number of the article
- `session`: (optional) the authentication credentials from `fs_auth`

#### Details
This function will make a draft or private article public, assigning it a DOI and making it permanently available through Figshare. If you use `fs_upload` to add new files to an existing public deposit, you must then use `fs_make_public` for those changes to be made in the public version of the repository.

#### Value
output of PUT request (invisibly)

#### Note
NOTE: Public articles are assigned DOIs and cannot be deleted or made private once declared public! Public articles do not count against your quota space.
Author(s)

Carl Boettiger <cboettig@gmail.com>

References

https://docs.figshare.com

See Also

fs_auth

Examples

## Not run:
fs_make_public(123)

## End(Not run)

fs_new_article Create a FigShare article.

Description

fs_new_article is a wrapper around many other rfigshare commands to provide convenient posting.

Usage

fs_new_article(
  title,
  description,
  type = c("dataset", "figure", "media", "poster", "paper", "fileset"),
  authors = NA,
  categories = NA,
  tags = NA,
  links = NA,
  files = NA,
  visibility = c("draft", "private", "public"),
  session = fs_get_auth()
)

Arguments

title for the article, see fs_create for details.
description of the article, see fs_create for details.
type one of: dataset, figure, media, poster, or paper, see fs_create for details.
authors Orded list of authors for the article, see fs_add_authors for details
**fs_search**

Advanced Search.

- **categories**: list of categories or category id numbers, see `fs_add_categories` for details.
- **tags**: list of tags, see `fs_add_tags` for details.
- **links**: list of links to add, see `fs_add_links` for details.
- **files**: path to the files to add, see `fs_upload` for details.
- **visibility**: one of "draft", "private" or "public". A draft document can still be edited and modified. A public document is visible to everyone and cannot be deleted (though additional authors to the work can still "claim" their authorship).
- **session**: (optional) credentials, see `link{fs_auth}`.

**Value**

- article id

**References**

[https://docs.figshare.com](https://docs.figshare.com)

**See Also**

- `fs_auth`, `fs_add_categories`, `fs_add_authors`, `fs_add_tags`, `fs_add_links`

**Examples**

```r
## Not run:
write.csv(mtcars, "mtcars.csv")
id <- fs_new_article(title="A Test of rfigshare",
                    description="This is a test of the fs_new_article function and related methods",
                    type="dataset",
                    authors=c("Karthik Ram", "Scott Chamberlain"),
                    tags=c("ecology", "openscience"),
                    categories="Ecology",
                    links="http://ropensci.org",
                    files="mtcars.csv",
                    visibility="private")

## End(Not run)
```

**Description**

Search function that will filter on matching timestamp, author, title, description, tag, category, and date range. Query searches against matches in any metadata field. Full-text searches coming soon.
Usage

fs_search(
  query, 
  author = NA, 
  title = NA, 
  description = NA, 
  tag = NA, 
  category = NA, 
  from_date = NA, 
  to_date = NA, 
  mine = FALSE, 
  public_only = FALSE, 
  private_only = FALSE, 
  drafts_only = FALSE, 
  session = fs_get_auth(), 
  base = "http://api.figshare.com/v1", 
  debug = FALSE 
)

Arguments

query the search query
author Show only results by this author
title Show only results matching or partially matching this title
description Show only results matching or partially matching this description
tag Show only results matching this tag
category Show only results matching this category
from_date Start time window for search. Date format is YYYY-MM-DD
to_date Ending time window for search. Date format is YYYY-MM-DD
mine Browse only articles owned by user. default is FALSE. Not functional. Use fs_browse instead.
public_only (for use with mine=TRUE only) browse only my public articles. default is FALSE
private_only (for use with mine=TRUE only) browse only my private articles. default is FALSE
drafts_only (for use with mine=TRUE only) browse only my draft articles. default is FALSE
session (optional) the authentication credentials from fs_auth. If not provided, will attempt to load from cache as long as figshare_auth has been run.
base the API access url
default logical, enable debugging mode

Value

output of PUT request (invisibly)
Author(s)  
Carl Boettiger <cboettig@gmail.com>  

References  
https://docs.figshare.com  

See Also  
\code{fs_auth} \code{fs_browse}  

Examples  
## Not run:  
fs_search("Boettiger")  
fs_search("Boettiger", author = "Carl")  
fs_search("Boettiger", author = "Carl", from="2014-01-01")  
fs_search("Boettiger", author = "Carl", from="2014-01-01",  
category = "Evolutionary Biology")  

## End(Not run)  

\begin{verbatim}  
fs_update  
\end{verbatim}

Update article title, description, or type  

Description  
Updates the article title, description or type. If any is not specified, it will remain unchanged.  

Usage  
fs_update(  
  article_id,  
  title = NA,  
  description = NA,  
  type = NA,  
  mine = TRUE,  
  session = fs_get_auth(),  
  debug = FALSE  
)  

Arguments  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>article_id</td>
<td>the id number of the article</td>
</tr>
<tr>
<td>title</td>
<td>for the article (to replace original title)</td>
</tr>
<tr>
<td>description</td>
<td>of the article (replaces original designation)</td>
</tr>
</tbody>
</table>
type one of: dataset, figure, media, poster, or paper (replaces original designation)
mine Set to TRUE if it refers to an item on your own account
session (optional) the authentication credentials from \texttt{fs_auth}.
debug return \texttt{httr} PUT request visibly?

Details

Updates the title, description, and type of an article.

Value

output of PUT request (invisibly)

Author(s)

Carl Boettiger \texttt{<cboettig@gmail.com>}

References

\url{https://docs.figshare.com}

See Also

\texttt{fs_auth}, \texttt{fs_add_tags}

Examples

```r
## Not run:
fs_update(138, title = "New title")
## End(Not run)
```

\texttt{fs_upload} \hspace{1cm} \textit{Upload file to an article}

Description

Upload file to an article

Usage

\texttt{fs_upload(article_id, file, session = fs_get_auth())}

Arguments

\begin{itemize}
\item \texttt{article_id} an article id number or a character string (or list) of numbers
\item \texttt{file} path to file to upload, or character string (or list) of files (paths)
\item \texttt{session} the authentication credentials from \texttt{fs_auth} (optional)
\end{itemize}
plot_to_filename

Details

Articles may be draft, private or public but all uploads are saved as draft changes - the canonical public version of the deposit is not updated. To update the public version of the repository, use \texttt{fs\_make\_public}. Only articles of type "fileset" can have multiple files uploaded.

If only a single id number is given but a character string of files is given, then be sure that the id corresponds to an object of type "fileset". If article\_id list has more than one id, then there must be a corresponding file path for each id.

Author(s)

Carl Boettiger <cboettig@gmail.com>

References

https://docs.figshare.com

See Also

\texttt{fs\_auth}

Examples

```r
## Not run:
id <- fs_create("Title", "description", "figure")
fs_upload(id, "file.png")

## End(Not run)
```

---

### plot_to_filename

Convienence function to save a ggplot2 plot, and return its filename.

#### Description

Convienence function to save a ggplot2 plot, and return its filename.

#### Usage

```r
plot_to_filename(plotobj, filename, path = ".")
```

#### Arguments

- `plotobj` ggplot2 plot object (should add support for base plots too)
- `filename` Filename, don’t include the file type extension.
- `path` Path where you want to save the file.

#### Value

A file name, to use in \texttt{fs\_upload}
See Also

fs_upload

Examples

## Not run:
# include in your fs_upload call
library(ggplot2)
p <- qplot(mpg, wt, data=mtcars)
plott <- fs_create(title="my title", description="some description", type="figure")
fs_add_categories(plott, "Ecology")
fs_upload(plott, plot_to_filename(p, "myfilename", "~"))

## End(Not run)

summary_fs_details Collect metadata from details call

Description

Collect metadata from details call

Usage

summary_fs_details(fs_details_obj)

Arguments

fs_details_obj object

Author(s)

Edmund Hart <edmund.m.hart@gmail.com>

References

https://docs.figshare.com

Examples

## Not run:
fs_auth()
my_article <- fs_details("138")
summary_fs_details(my_article)

## End(Not run)
Index

download.file, 14
fs_add_authors, 2, 9, 18, 19
fs_add_categories, 3, 9, 19
fs_add_links, 4, 19
fs_add_tags, 9, 19, 22
fs_auth, 2–5, 5, 6–14, 16–23
fs_author_search, 6
fs_browse, 7, 20, 21
fs_cat_to_id, 9
fs_category_list, 8
fs_create, 9, 9, 18
fs_create_author, 9, 10
fs_delete, 11
fs_details, 12
fs_download, 13
fs_embed, 14
fs_ids, 15
fs_image_url, 16
fs_make_private, 9, 16
fs_make_public, 9, 17, 23
fs_new_article, 18
fs_search, 19
fs_update, 9, 21
fs_upload, 9, 10, 17, 19, 22, 24

plot_to_filename, 23
summary_fs_details, 24