

Package ‘rdryad’

June 18, 2018

Type Package

Title Access for Dryad Web Services

Description Interface to the Dryad “Solr” API, their “OAI-PMH” service, and fetch datasets. Dryad (<<http://datadryad.org/>>) is a curated host of data underlying scientific publications.

Version 0.4.0

License MIT + file LICENSE

URL <https://github.com/ropensci/rdryad>

BugReports <https://github.com/ropensci/rdryad/issues>

Imports crul (>= 0.4.0), curl (>= 3.0), xml2 (>= 1.0.0), oai (>= 0.2.2), solrium (>= 1.0.0), data.table, tibble

Suggests testthat

RoxygenNote 6.0.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre] (<<https://orcid.org/0000-0003-1444-9135>>),
Carl Boettiger [aut] (<<https://orcid.org/0000-0002-1642-628X>>),
Karthik Ram [ctb]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2018-06-18 16:44:52 UTC

R topics documented:

rdryad-package	2
doi2handle	3
dryad_fetch	3
dryad_files	4
dryad_metadata	5
dryad_package_dois	6
dr_get_records	7

dr_identify	7
dr_list_identifiers	8
dr_list_metadata_formats	9
dr_list_records	10
dr_list_sets	11
d_solr_search	11

Index	14
--------------	-----------

rdryad-package	<i>Interface to the Dryad Web services</i>
----------------	--

Description

Includes access to Dryad's Solr API, OAI-PMH service, and part of their REST API.

Package API

The following functions work with the Dryad Solr service

- [d_solr_facet\(\)](#)
- [d_solr_group\(\)](#)
- [d_solr_highlight\(\)](#)
- [d_solr_mlt\(\)](#)
- [d_solr_search\(\)](#)
- [d_solr_stats\(\)](#)

The following functions work with the Dryad OAI-PMH service

- [dr_get_records\(\)](#)
- [dr_identify\(\)](#)
- [dr_list_identifiers\(\)](#)
- [dr_list_metadata_formats\(\)](#)
- [dr_list_records\(\)](#)
- [dr_list_sets\(\)](#)

The following functions sort out file URLs and help you download those files

- [dryad_fetch\(\)](#)
- [dryad_files\(\)](#)
- [dryad_metadata\(\)](#)
- [dryad_package_dois\(\)](#)

These functions convert between Dryad handles and DOIs

- [handle2doi\(\)](#)
- [doi2handle\(\)](#)

Author(s)

Scott Chamberlain <myrmecocystus@gmail.com>

doi2handle *Get a Dryad DOI from a handle, and vice versa*

Description

Get a Dryad DOI from a handle, and vice versa

Usage

```
doi2handle(x, ...)
```

```
handle2doi(x, ...)
```

Arguments

x (character) A Dryad dataset DOI or handle. required
... Curl options, passed on to [curl::HttpClient](#)

Value

(character) a DOI or handle

Examples

```
## Not run:  
doi2handle('10.5061/dryad.c0765')  
handle2doi('10255/dryad.153920')  
doi2handle('10.5061/dryad.c0765')  
  
## End(Not run)
```

dryad_fetch *Download Dryad files*

Description

Download Dryad files

Usage

```
dryad_fetch(url, destfile = NULL, try_file_names = FALSE, ...)
```

Arguments

url (character) One or more Dryad URL for a dataset

destfile (character) Destination file. If not given, we assign a file name based on URL provided.

try_file_names (logical) try to parse file names out of the URLs. Default: FALSE

... Further args passed on to `curl::curl_download()`

Details

This function is a thin wrapper around `curl::curl_download()` to get files to your machine only. We don't attempt to read/parse them

Value

named (list) with path(s) to the file(s) - list names are the urls passed into the url parameter

Examples

```
## Not run:
# Single file
x <- dryad_files('10.5061/dryad.1758')

## without specifying a destination file
dryad_fetch(url = x)

## specify a destination file
dryad_fetch(url = x[1], (f <- tempfile(fileext = ".csv")))

## use try_file_names - we try to extract file names from URLs
dryad_fetch(url = x, try_file_names = TRUE)

# Many files
x <- dryad_files(doi = '10.5061/dryad.60699')
res <- dryad_fetch(x)
head(read.delim(res[[1]], sep = ";"))

## End(Not run)
```

dryad_files

Get a URL given a Dryad DOI

Description

To get a DOI from a Dryad Handle, use `handle2doi()`

Usage

```
dryad_files(doi, ...)
```

Arguments

doi (character) A Dryad dataset DOI, of the form 10.5061/dryad.xxx. required
 ... Curl options, passed on to [crul::HttpClient](#)

Value

(character) One or more URLs for direct download of datasets for the given Dryad DOI

Examples

```
## Not run:
dryad_files(doi = '10.5061/dryad.1758')
dryad_files(doi = '10.5061/dryad.60699')

# if you have a handle, use handle2doi() to convert to a DOI
(doi <- handle2doi('10255/dryad.153920'))
(files <- dryad_files(doi))
(out <- dryad_fetch(files))
# file sizes in MB
vapply(out, function(x) file.info(x)[["size"]], 1) / 10^6

## End(Not run)
```

dryad_metadata	<i>Download Dryad file metadata</i>
----------------	-------------------------------------

Description

Download Dryad file metadata

Usage

```
dryad_metadata(doi, ...)
```

Arguments

doi (character) A Dryad DOI for a dataset of files within a dataset
 ... Further args passed on to [crul::HttpClient](#)

Value

named (list) with slots for:

- desc: object metadata
- files: file information
- attributes: metadata about the metadata file
- structMap: not sure what this is

Examples

```
## Not run:
dryad_metadata('10.5061/dryad.1758')
dryad_metadata('10.5061/dryad.9t0n8/1')
dryad_metadata('10.5061/dryad.60699/3')
out <- dryad_metadata('10.5061/dryad.60699/5')
out$desc$text[out$desc$qualifier %in% c("pageviews", "downloads")]

## End(Not run)
```

`dryad_package_dois` *Get file DOIs for a Dryad package DOI*

Description

Get file DOIs for a Dryad package DOI

Usage

```
dryad_package_dois(doi, ...)
```

Arguments

`doi` (character) A Dryad package DOI. required
`...` Further args passed on to [crul::HttpClient](#)

Value

(character) zero or more DOIs for the files; if no results a zero length character vector

Examples

```
## Not run:
dryad_package_dois('10.5061/dryad.1758')
dryad_package_dois('10.5061/dryad.9t0n8')
dryad_package_dois('10.5061/dryad.60699')

## End(Not run)
```

dr_get_records	<i>Download metadata for individual Dryad id's</i>
----------------	--

Description

Download metadata for individual Dryad id's

Usage

```
dr_get_records(ids, prefix = "oai_dc", as = "df", ...)
```

Arguments

ids	Dryad identifier, i.e. oai:datadryad.org:10255/dryad.8820
prefix	A character string to specify the metadata format in OAI-PMH requests issued to the repository. The default ("oai_dc") corresponds to the mandatory OAI unqualified Dublin Core metadata schema.
as	(character) What to return. One of "df" (for data.frame; default), "list", or "raw" (raw text)
...	Curl debugging options passed on to <code>httr::GET</code>

Value

XML character string, data.frame, or list, depending on what requested with the as parameter

Examples

```
## Not run:
dr_get_records(ids = 'oai:datadryad.org:10255/dryad.8820')
handles <- c('10255/dryad.36217', '10255/dryad.86943', '10255/dryad.84720',
            '10255/dryad.34100')
ids <- paste0('oai:datadryad.org:', handles)
dr_get_records(ids)

## End(Not run)
```

dr_identify	<i>Learn about the Dryad OAI-PMH service.</i>
-------------	---

Description

Learn about the Dryad OAI-PMH service.

Usage

```
dr_identify(...)
```

Arguments

... Curl debugging options passed on to `httr::GET`

Value

List of information describing Dryad.

Examples

```
## Not run:
dr_identify()

## End(Not run)
```

`dr_list_identifiers` *Gets OAI Dryad identifiers*

Description

Gets OAI Dryad identifiers

Usage

```
dr_list_identifiers(prefix = "oai_dc", from = NULL, until = NULL,
  set = "hdl_10255_3", token = NULL, as = "df", ...)
```

Arguments

<code>prefix</code>	A character string to specify the metadata format in OAI-PMH requests issued to the repository. The default ("oai_dc") corresponds to the mandatory OAI unqualified Dublin Core metadata schema.
<code>from</code>	Character string giving datestamp to be used as lower bound for datestamp-based selective harvesting (i.e., only harvest records with datestamps in the given range). Dates and times must be encoded using ISO 8601. The trailing Z must be used when including time. OAI-PMH implies UTC for data/time specifications.
<code>until</code>	Character string giving a datestamp to be used as an upper bound, for datestamp-based selective harvesting (i.e., only harvest records with datestamps in the given range).
<code>set</code>	A character string giving a set to be used for selective harvesting (i.e., only harvest records in the given set).
<code>token</code>	(character) a token previously provided by the server to resume a request where it last left off. 50 is max number of records returned. We will loop for you internally to get all the records you asked for.
<code>as</code>	(character) What to return. One of "df" (for data.frame; default), "list", or "raw" (raw text)
...	Curl debugging options passed on to <code>httr::GET</code>

Value

XML character string, data.frame, or list, depending on what requested with the as parameter
List of OAI identifiers for each dataset.

Examples

```
## Not run:  
dr_list_identifiers(from='2010-01-01', until = "2010-06-30")  
dr_list_identifiers(prefix="mets", from='2015-09-01', until='2015-09-20')  
  
identifiers <- dr_list_identifiers('rdf')  
  
# Data packages  
identifiers[[1]]  
  
# Data files  
identifiers[[2]]  
  
## End(Not run)
```

dr_list_metadata_formats

Get available Dryad metadata formats

Description

Get available Dryad metadata formats

Usage

```
dr_list_metadata_formats(...)
```

Arguments

... Curl debugging options passed on to httr::GET

Value

List of information on metadata formats.

Examples

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

dr_list_records *List Dryad records*

Description

List Dryad records

Usage

```
dr_list_records(prefix = "oai_dc", from = NULL, until = NULL,
  set = "hdl_10255_3", token = NULL, as = "df", ...)
```

Arguments

prefix	A character string to specify the metadata format in OAI-PMH requests issued to the repository. The default ("oai_dc") corresponds to the mandatory OAI unqualified Dublin Core metadata schema.
from	Character string giving datestamp to be used as lower bound for datestamp-based selective harvesting (i.e., only harvest records with datestamps in the given range). Dates and times must be encoded using ISO 8601. The trailing Z must be used when including time. OAI-PMH implies UTC for data/time specifications.
until	Character string giving a datestamp to be used as an upper bound, for datestamp-based selective harvesting (i.e., only harvest records with datestamps in the given range).
set	A character string giving a set to be used for selective harvesting (i.e., only harvest records in the given set).
token	(character) a token previously provided by the server to resume a request where it last left off. 50 is max number of records returned. We will loop for you internally to get all the records you asked for.
as	(character) What to return. One of "df" (for data.frame; default), "list", or "raw" (raw text)
...	Curl debugging options passed on to <code>httr::GET</code>

Value

XML character string, data.frame, or list, depending on what requested with the as parameter

Examples

```
## Not run:
dr_list_records(from='2016-01-01', until='2016-09-10')

## End(Not run)
```

dr_list_sets	<i>List the sets in the Dryad metadata repository.</i>
--------------	--

Description

Retrieve the set structure of Dryad, useful for selective harvesting

Usage

```
dr_list_sets(token = NULL, as = "df", ...)
```

Arguments

token	(character) a token previously provided by the server to resume a request where it last left off. 50 is max number of records returned. We will loop for you internally to get all the records you asked for.
as	(character) What to return. One of "df" (for data.frame; default), "list", or "raw" (raw text)
...	Curl debugging options passed on to <code>httr::GET</code>

Examples

```
## Not run:  
dr_list_sets()  
dr_list_sets(as = "list")  
dr_list_sets(as = "raw")  
  
## End(Not run)
```

d_solr_search	<i>Search the Dryad Solr endpoint.</i>
---------------	--

Description

Search the Dryad Solr endpoint.

Usage

```
d_solr_search(..., proxy = NULL, callopts = list())  
d_solr_facet(..., proxy = NULL, callopts = list())  
d_solr_group(..., proxy = NULL, callopts = list())  
d_solr_highlight(..., proxy = NULL, callopts = list())
```

```
d_solr_mlt(..., proxy = NULL, callopts = list())
```

```
d_solr_stats(..., proxy = NULL, callopts = list())
```

Arguments

...	Solr parameters passed on to the respective solrium package function.
proxy	List of arguments for a proxy connection, including one or more of: url, user, pwd, and auth. See crul::proxy for help, which is used to construct the proxy connection.
callopts	Further args passed on to crul::HttpClient

Details

See the **solrium** package documentation for available parameters. For each of `d_solr_search`, `d_solr_facet`, `d_solr_stats`, and `d_solr_mlt`, `d_solr_group`, and `d_solr_highlight` see the equivalently named function in **solrium**.

The `wt` parameter is now hard-coded to `xml` because a recent change in the Dryad Solr infrastructure makes it impossible to get JSON output - this shouldn't affect most users. In addition, we hard code a `curl` option to follow redirects, just so you're aware.

Examples

```
## Not run:
# Basic search
d_solr_search(q="Galliard")

# Basic search, restricting to certain fields
d_solr_search(q="Galliard", fl=c('handle', 'dc.title_sort'))

# Search all text for a string, but limits results to two specified fields:
d_solr_search(q="dwc.ScientificName:drosophila", fl='handle,dc.title_sort')

# Dryad data based on an article DOI:
d_solr_search(q="dc.relation.isreferencedby:10.1038/nature04863",
  fl="dc.identifier,dc.title_ac")

# All terms in the dc.subject facet, along with their frequencies:
d_solr_facet(q="location:l2", facet.field="dc.subject_filter", facet.minCount=1,
  facet.limit=10)

# Article DOIs associated with all data published in Dryad over the past 90 days:
d_solr_search(q="dc.date.available_dt:[NOW-90DAY/DAY TO NOW]",
  fl="dc.relation.isreferencedby", rows=10)

# Data DOIs published in Dryad during January 2011
query <- "location:l2 dc.date.available_dt:[2011-01-01T00:00:00Z TO 2011-01-31T23:59:59Z]"
d_solr_search(q=query, fl="dc.identifier", rows=10)
```

```
# Highlight
d_solr_highlight(q="bird", hl.fl="dc.description")

# More like this
d_solr_mlt(q="bird", mlt.count=10, mlt.fl='dc.title_sort', fl='handle,dc.title_sort')

# Stats
d_solr_stats(q="*:*", stats.field="dc.date.accessioned.year")

## End(Not run)
```

Index

`crul::HttpClient`, 3, 5, 6, 12
`crul::proxy`, 12
`curl::curl_download()`, 4

`d_solr_facet(d_solr_search)`, 11
`d_solr_facet()`, 2
`d_solr_group(d_solr_search)`, 11
`d_solr_group()`, 2
`d_solr_highlight(d_solr_search)`, 11
`d_solr_highlight()`, 2
`d_solr_mlt(d_solr_search)`, 11
`d_solr_mlt()`, 2
`d_solr_search`, 11
`d_solr_search()`, 2
`d_solr_stats(d_solr_search)`, 11
`d_solr_stats()`, 2
`doi2handle`, 3
`doi2handle()`, 2
`dr_get_records`, 7
`dr_get_records()`, 2
`dr_identify`, 7
`dr_identify()`, 2
`dr_list_identifiers`, 8
`dr_list_identifiers()`, 2
`dr_list_metadata_formats`, 9
`dr_list_metadata_formats()`, 2
`dr_list_records`, 10
`dr_list_records()`, 2
`dr_list_sets`, 11
`dr_list_sets()`, 2
`dryad_fetch`, 3
`dryad_fetch()`, 2
`dryad_files`, 4
`dryad_files()`, 2
`dryad_metadata`, 5
`dryad_metadata()`, 2
`dryad_package_dois`, 6
`dryad_package_dois()`, 2

`handle2doi(doi2handle)`, 3

`handle2doi()`, 2, 4
`rdryad(rdryad-package)`, 2
`rdryad-package`, 2