Package ‘rdatacite’

May 8, 2019

Type Package

Title 'DataCite' Client for 'OAI-PMH' Methods and their Search API

Description Client for the web service methods provided by 'DataCite' (<https://www.datacite.org/>), including functions to interface with their 'OAI-PMH' 'metadata' service, and a 'RESTful' search API. The API is backed by 'SOLR', allowing expressive queries, including faceting, statistics on variables, and 'more-like-this' queries.

Version 0.4.2

License MIT + file LICENSE

URL https://github.com/ropensci/rdatacite

BugReports https://github.com/ropensci/rdatacite/issues

Encoding UTF-8

Imports oai (>= 0.2.2), solrium (>= 1.0.0), jsonlite, crul (>= 0.7.4)

Suggests knitr, testthat, webmockr, vcr

RoxygenNote 6.1.1

X-schema.org-applicationCategory Data

X-schema.org-keywords data, scholarly, dataset, https, API, web-services, solr

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

NeedsCompilation no

Author Scott Chamberlain [aut, cre] (<https://orcid.org/0000-0003-1444-9135>)

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2019-05-07 22:20:03 UTC

R topics documented:

rdatacite-package ........................................................................................................ 2
dc_data_center ........................................................................................................... 3
DataCite R client.

Description

DataCite R client.

OAI-PMH functions

- `dc_oai_getrecord()` - Get records
- `dc_oai_identify()` - Identify the OAI-PMH service
- `dc_oai_listidentifiers()` - List identifiers
- `dc_oai_listmetadataformats()` - List metadata formats
- `dc_oai_listrecords()` - List records
- `dc_oai_listsets()` - List sets

Search functions (Solr based)

- `dc_search()` - General search, only returns documents
- `dc_facet()` - Faceting only (w/o general search)
- `dc_mlt()` - More like this (w/o general search)
- `dc_stats()` - Stats search (w/o general search)

REST API functions

- `dc_data_center()` - Get a single data center
- `dc_data_centers()` - Search data centers
- `dc_member()` - Get single DataCite member
- `dc_members()` - Search DataCite members
- `dc_work()` - Get single works
- `dc_works()` - Search works
dc_data_center

Vignettes

Coming soon...

Author(s)

Scott Chamberlain <myrmecocystus@gmail.com>

dc_data_center  DataCite REST API: data center

Description

DataCite REST API: data center

Usage

dc_data_center(data_center_id, include = NULL, ...)

Arguments

data_center_id  (character) a data center id
include         (character) vector of member fields to return
...             curl options passed on to curl::HttpClient

See Also

dc_data_centers()

Other rest api methods: dc_data_centers, dc_members, dc_member, dc_works, dc_work

Examples

## Not run:
# dc_data_center("delft.data4t")

## End(Not run)
dc_data_centers  

DataCite REST API: data centers

Description

DataCite REST API: data centers

Usage

```r
dc_data_centers(query = NULL, member_id = NULL, offset = 0,
rows = 25, include = NULL, ...)
```

Arguments

- `query` (character) Query string, e.g., data center name
- `member_id` (character) Member that the data center is associated with
- `offset` (numeric/integer) result offset, the record to start at
- `rows` (numeric/integer) results per page
- `include` (character) vector of member fields to return
- `...` curl options passed on to `crul::HttpClient`

See Also

- `dc_data_center`

Other rest api methods: `dc_data_center`, `dc_members`, `dc_member`, `dc_works`, `dc_work`

Examples

```r
## Not run:
dc_data_centers(member_id = "delft")
dc_data_centers(member_id = "delft", include = "member")

## End(Not run)
```

dc_member  

DataCite REST API: member

Description

DataCite REST API: member

Usage

```r
dc_member(member_id, ...)
```
**dc_members**

**Arguments**

- **member_id** (character) Member that the data center is associated with
- ... curl options passed on to **cru::HttpClient**

**See Also**

- **dc_members()**

Other rest api methods: **dc_data_centers, dc_data_center, dc_members, dc_works, dc_work**

**Examples**

```r
## Not run:
dc_member("cern")

## End(Not run)
```

---

**dc_members**

*DataCite REST API: members*

**Description**

DataCite REST API: members

**Usage**

```r
dc_members(query = NULL, member_type = NULL, region = NULL,
         year = NULL, offset = 0, rows = 25, ...)
```

**Arguments**

- **query** (character) Query string, e.g., data center name
- **member_type** (character) member type, one of "allocating" or "non-allocating"
- **region** (character) Geographic region, one of "amer", "apac", or "emea"
- **year** (numeric/integer) year member joined DataCite
- **offset** (numeric/integer) result offset, the record to start at
- **rows** (numeric/integer) results per page
- ... curl options passed on to **cru::HttpClient**

**See Also**

- **dc_member()**

Other rest api methods: **dc_data_centers, dc_data_center, dc_member, dc_works, dc_work**
dc_oai_getrecord

Examples

## Not run:
dc_members(query = "ands")

## End(Not run)

dc_oai_getrecord  Retrieve an individual record from the DataCite repository

Description

Retrieve an individual record from the DataCite repository

Usage

dc_oai_getrecord(id)

Arguments

id  DataCite identifier, e.g., "oai:oai.datacite.org:6718729". If left blank, get all metadata formats

Examples

## Not run:
# you can pass in just the numeric part of the ID
dc_oai_getrecord(56225)
dc_oai_getrecord(c(56225, 6667400))

# Or, the entire thing
dc_oai_getrecord("oai:oai.datacite.org:56225")

# Or, mixed
dc_oai_getrecord(c("56225", "oai:oai.datacite.org:6667400"))

today <- format(Sys.Date(), "%Y-%m-%d")
temp <- dc_oai_listidentifiers(from = today)
dc_oai_getrecord(temp$identifier[1:2])

## End(Not run)
### dc_oai_identify

**Learn about the Dryad OAI-PMH service**

**Description**

Learn about the Dryad OAI-PMH service

**Usage**

```r
dc_oai_identify(...)```

**Arguments**

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

**Examples**

```r
library(curl)
url <- "https://oai.datacite.org/oai"
if (curl::HttpClient$new(url=url)$get()$success()) {
  dc_oai_identify()
}
```

### dc_oai_listidentifiers

**Retrieve record headers**

**Description**

Retrieve record headers

**Usage**

```r
dc_oai_listidentifiers(from = NULL, until = NULL, set = NULL, prefix = "oai_dc", token = NULL, ...)
```

**Arguments**

- `from`: specifies that records returned must have been created/update/deleted on or after this date.
- `until`: specifies that records returned must have been created/update/deleted or before this date.
- `set`: specifies the set that returned records must belong to.
- `prefix`: specifies the metadata format that the records will be returned in. One of: `oai_dc` (default), `oai_datacite`, or `datacite`. See Prefixes for more info.
dc_oai_listmetadataformats

- **token**: a token previously provided by the server to resume a request where it last left off.
- **...**: Curl options passed on to `http::GET`

**Examples**

```r
## Not run:
today <- format(Sys.Date(), "%Y-%m-%d")
dc_oai_listidentifiers(from = today)
dc_oai_listidentifiers(from = '2011-06-01T', until = '2011-07-01T')
dc_oai_listidentifiers(set = "REFQUALITY")

## End(Not run)
```

---

**dc_oai_listmetadataformats**

*List available metadata formats that the DataCite repository can disseminate*

**Description**

List available metadata formats that the DataCite repository can disseminate

**Usage**

```r
dc_oai_listmetadataformats(id = NULL, ...)
```

**Arguments**

- **id**: DataCite identifier, e.g., "oai:oai.datacite.org:6718729". If left blank, get all metadata formats.
- **...**: Curl options passed on to `http::GET`

**Examples**

```r
## Not run:
dc_oai_listmetadataformats()
dc_oai_listmetadataformats("oai:oai.datacite.org:6718729")

## End(Not run)
```
dc_oai_listrecords

List the records in the DataCite metadata repository.

Description
List the records in the DataCite metadata repository.

Usage

dc_oai_listrecords(from = NULL, until = NULL, set = NULL,
prefix = "oai_dc", token = NULL, ...)

Arguments

from specifies that records returned must have been created/update/deleted on or after
this date.
until specifies that records returned must have been created/update/deleted or before
this date.
set specifies the set that returned records must belong to.
prefix specifies the metadata format that the records will be returned in. One of: oai_dc
(default), oai_datacite, or datacite. See Prefixes for more info.
token a token previously provided by the server to resume a request where it last left
off.
... Curl options passed on to http::GET

Examples

## Not run:
dc_oai_listrecords(from = '2011-06-01T', until = '2011-07-01T')
dc_oai_listrecords(from = '2011-06-01T', until = '2011-11-01T', set = "REFQUALITY")

## End(Not run)

dc_oai_listsets

Retrieve the set structure of DataCite

Description
Useful for selective harvesting

Usage

dc_oai_listsets(token = NULL, ...)
**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>token</td>
<td>a token previously provided by the server to resume a request where it last left off.</td>
</tr>
<tr>
<td>...</td>
<td>Curl options passed on to <code>httr::GET</code></td>
</tr>
</tbody>
</table>

**Examples**

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

---

**dc_search** | *DataCite Solr web service for search, facet, stats, and mlt queries*

**Description**

DataCite Solr web service for search, facet, stats, and mlt queries

**Usage**

```r
dc_search(..., proxy = NULL, callopts = list())
dc_facet(..., proxy = NULL, callopts = list())
dc_stats(..., proxy = NULL, callopts = list())
dc_mlt(..., proxy = NULL, callopts = list())
```

**Arguments**

```r
...
proxy
  List of arguments for a proxy connection, including one or more of: url, port, username, password, and auth. See `proxy` for help, which is used to construct the proxy connection.
callopts
  Curl options in a named list passed on to `HttpClient`, see `curl-options`
```

**Details**

See the `solrium` package documentation for available parameters. For each of `dc_search`, `dc_facet`, `dc_stats`, and `dc_mlt` see the equivalently named function in `solrium`:

- `dc_search` - `solr_search`
- `dc_facet` - `solr_facet`
- `dc_stats` - `solr_stats`
Fields you can search on and retrieve back are:

- doi
- creator
- publisher
- publicationYear
- title
- alternateIdentifier
- contributor
- date
- description
- format
- language
- relatedIdentifier
- resourceType
- resourceTypeGeneral
- rights
- size
- subject
- version
- allocator symbol of the DataCite member (e.g. BL = British Library)
- datacentre symbol of the datacentre, which uploaded the metadata (e.g. BL.ADS = Archaeology Data Service)
- indexed datestamp of indexing
- prefix DOI prefix (e.g. 10.5284)
- refQuality reference quality flag (1 or 0)
- uploaded datestamp of metadata uploading

If you want to use wt='csv' you’ll need solrium package version 0.1.8.99 or greater. Installation instructions at https://github.com/ropensci/solrium#quick-start.

References

https://support.datacite.org/docs/datacite-search-user-documentation
Examples

library(crlul)
url <- "https://search.datacite.org/api"
if (crlul::HttpClient$new(url=url)$get()$success()) {

  # Search
  ## various searches
  ### containing laser
  dc_search(q = "laser", fl=c('doi','publicationYear'), rows=5)

  ### containing laser and safety
  dc_search(q = "laser safety", fl=c('doi','publicationYear'),
            rows=5)

  ### containing laser and not medicine
  dc_search(q = "laser -medicine", fl=c('doi',
                                       'publicationYear'), rows=5)

  
  # Not run:
  ## containing laser and red or green
  dc_search(q = "laser AND (red OR green)", fl=c('doi',
                                               'publicationYear'), rows=5)

  ### containing wind turbine as a phrase in the title
  dc_search(q = 'title:"wind turbine"', fl=c('doi','title'),
            rows=5)

  ### containing geology in metadata field subject
  dc_search(q = "subject:geology", fl=c('doi','subject'),
            rows=5)

  ### belonging to DataCite members BL or TIB
  dc_search(q = "allocator:(BL OR TIB)" , fl=c('doi','allocator'),
            rows=5)

  ### published between 2000 and 2005
  dc_search(q = "publicationYear:[2000 TO 2005]",
            fl=c('doi','publicationYear'), rows=5)

  ### uploaded to DataCite in the last 5 days
  dc_search(q = "uploaded:[NOW-5DAYS TO NOW]",
            fl=c('doi','uploaded'), rows=5)

  # search with csv output
  ## If you want to use wt=csv, install development version of
  ## solr package, see Details
  dc_search(q = 'wind', fl=c('doi','title'), wt='csv')

  # Faceting
  dc_facet(q = "wind", facet.field='publisher_facet')
dc_work

# Stats
dc_stats(q = "ecology", stats.field='date')

# More like this (aka mlt)
dc_mlt(q = "ecology", mlt.fl='title', mlt.count=5,
     fl=c('doi','title'))

## End(Not run)

---

**dc_work**

**DataCite REST API: single works**

---

**Description**

DataCite REST API: single works

**Usage**

dc_work(doi, ...)

**Arguments**

<table>
<thead>
<tr>
<th>doi</th>
<th>(character) a DOI</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>curl options passed on to <code>crul::HttpClient</code></td>
</tr>
</tbody>
</table>

**See Also**

dc_works()

dc_data_centers, dc_data_center, dc_members, dc_member, dc_works

**Examples**

```r
## Not run:
dc_work(doi = "10.5438/0012")

## End(Not run)
```
dc_works  

DataCite REST API: works

Description

DataCite REST API: works

Usage

dc_works(query = NULL, order = NULL, sort = NULL, rows = NULL,  
onset = NULL, include = NULL, ...)

Arguments

query   (character) Query string, e.g., any part of a work
order   (character) set the sort order to one of "asc" or "desc"
sort    (character) sort results by a certain field, one of "deposited", "published", "updated" or "score"
rows    (numeric/integer) results per page
offset   (numeric/integer) result offset, the record to start at
include  (character) vector of member fields to return
...  curl options passed on to crul::HttpClient

See Also

dc_work()

Other rest api methods: dc_data_centers, dc_data_center, dc_members, dc_member, dc_work

Examples

## Not run:
dc_works(query = "climate change")

## End(Not run)
**Prefixes**

**Metadata Prefixes**

---

**Description**

These three options can be used in `dc_oai_*( )` functions as input to the `prefix` parameter.

**Details**

- **oai_dc** - OAI Dublin Core - As a minimum requirement for OAI-PMH compliance, metadata must be made available in the OAI Dublin Core format. For more information please see the OAI-PMH web site.

- **oai_datacite** - OAI DataCite - This metadata format has been specifically established for the dissemination of DataCite records using OAI-PMH. In addition to the original DataCite metadata, this format contains several other elements describing the version of the metadata, whether it is of reference quality, and the registering datacentre. For more information about this format and its schema please see the Datacite OAI schema web site.

- **datacite** - DataCite Direct - This metadata format contains only the original DataCite metadata without additions or alterations. Because there are multiple versions of DataCite metadata in the MDS, there is no one schema that they will all adhere to. Therefore the schema for this format does not exist and metadata will not validate against it. Please note that this format is not OAI-PMH version 2.0 compliant for the previously stated reasons.
Index

*Topic package
   rdatacite-package, 2
   crul::HttpClient, 3–5, 13, 14
   dc_data_center, 3, 4, 5, 13, 14
dc_data_center(), 2, 4
dc_data_centers, 3, 4, 5, 13, 14
dc_data_centers(), 2, 3
dc_facet (dc_search), 10
dc_facet(), 2
dc_member, 3, 4, 4, 5, 13, 14
dc_member(), 2, 5
dc_members, 3–5, 5, 13, 14
dc_members(), 2, 5
dc_mlt (dc_search), 10
dc_mlt(), 2
dc_oai_getrecord, 6
dc_oai_getrecord(), 2
dc_oai_identify, 7
dc_oai_identify(), 2
dc_oai_listidentifiers, 7
dc_oai_listidentifiers(), 2
dc_oai_listmetadataformats, 8
dc_oai_listmetadataformats(), 2
dc_oai_listrecords, 9
dc_oai_listrecords(), 2
dc_oai_listsets, 9
dc_oai_listsets(), 2
dc_search, 10
dc_search(), 2
dc_stats (dc_search), 10
dc_stats(), 2
dc_work, 3–5, 13, 14
dc_work(), 2, 14
dc_works, 3–5, 13, 14
dc_works(), 2, 13
   HttpClient, 10
   Prefixes, 15

proxy, 10
rdatacite (rdatacite-package), 2
rdatacite-package, 2
   solr_facet, 10
   solr_mlt, 11
   solr_search, 10
   solr_stats, 10