

Package: pangaeear (via r-universe)

December 16, 2024

Title Client for the 'Pangaea' Database

Description Tools to interact with the 'Pangaea' Database (<<https://www.pangaea.de>>), including functions for searching for data, fetching 'datasets' by 'dataset' 'ID', and working with the 'Pangaea' 'OAI-PMH' service.

Version 1.1.0

License MIT + file LICENSE

URL <https://github.com/ropensci/pangaeear> (devel),
<https://docs.ropensci.org/pangaeear/> (documentation)

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

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Imports crul (>= 0.4.0), jsonlite (>= 1.5), xml2 (>= 1.1.1), oai (>= 0.2.2), tibble (>= 1.1), hoardr (>= 0.2.0), png

Suggests knitr, testthat, vcr, markdown

RoxygenNote 7.1.1

X-schema.org-applicationCategory Biodiversity

X-schema.org-keywords Pangaea, environmental science, earth science, archive, paleontology, ecology, chemistry, atmosphere

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

Config/pak/sysreqs libicu-dev libpng-dev libxml2-dev libssl-dev

Repository <https://ropensci.r-universe.dev>

RemoteUrl <https://github.com/ropensci/pangaeear>

RemoteRef master

RemoteSha 98de95524ea10e21142571ddb9b16989f64fb8c

Contents

| | |
|------------------------------------|-----------|
| pangaeear-package | 2 |
| pg_cache | 2 |
| pg_cache_clear | 3 |
| pg_cache_list | 4 |
| pg_data | 4 |
| pg_get_record | 6 |
| pg_identify | 7 |
| pg_list_identifiers | 8 |
| pg_list_metadata_formats | 9 |
| pg_list_records | 10 |
| pg_list_sets | 11 |
| pg_search | 12 |
| pg_search_es | 14 |
| Index | 17 |

| | |
|-------------------|--|
| pangaeear-package | <i>Client for the Pangaea Database</i> |
|-------------------|--|

Description

Pangaea database

Details

Package includes tools to interact with the Pangaea Database, including functions for searching for data, fetching datasets by dataset ID, working with the Pangaea OAI-PMH service, and Elastic-search service.

Getting data

The main workhorse function for getting data is `pg_data()`. One thing you may want to do is set a different path for caching the data you download: see `pg_cache` for details

| | |
|----------|----------------|
| pg_cache | <i>Caching</i> |
|----------|----------------|

Description

Manage cached pangaeear files with **hoardr**

Details

The default cache directory is `paste0(rappdirs::user_cache_dir(), "/R/pangaeear")`, but you can set your own path using `cache_path_set()`

`cache_delete` only accepts 1 file name, while `cache_delete_all` doesn't accept any names, but deletes all files. For deleting many specific files, use `cache_delete` in a `lapply()` type call

Useful user functions

- `pg_cache$cache_path_get()` get cache path
- `pg_cache$cache_path_set()` set cache path
- `pg_cache$list()` returns a character vector of full path file names
- `pg_cache$files()` returns file objects with metadata
- `pg_cache$details()` returns files with details
- `pg_cache$delete()` delete specific files
- `pg_cache$delete_all()` delete all files, returns nothing

Examples

```
## Not run:
pg_cache

# list files in cache
pg_cache$list()

# delete certain database files
# pg_cache$delete("file path")
# pg_cache$list()

# delete all files in cache
# pg_cache$delete_all()
# pg_cache$list()

# set a different cache path from the default
# pg_cache$cache_path_set(full_path = "/Foo/Bar")

## End(Not run)
```

| | |
|-----------------------------|-------------------------|
| <code>pg_cache_clear</code> | <i>cache path clear</i> |
|-----------------------------|-------------------------|

Description

cache path clear

Usage

```
pg_cache_clear(...)
```

Arguments

... ignored

pg_cache_list *cache list*

Description

cache list

Usage

pg_cache_list(...)

Arguments

... ignored

pg_data *Download data from Pangaea.*

Description

Grabs data as a dataframe or list of dataframes from a Pangaea data repository URI; see: <https://www.pangaea.de/>

Usage

pg_data(doi, overwrite = TRUE, mssgs = TRUE, ...)

Arguments

doi DOI of Pangaea single dataset, or of a collection of datasets. Expects either just a DOI of the form 10.1594/PANGAEA.746398, or with the URL part in front, like <https://doi.pangaea.de/10.1594/PANGAEA.746398>

overwrite (logical) Overwrite a file if one is found with the same name

mssgs (logical) print information messages. Default: TRUE

... Curl options passed on to [curl::verb-GET](#)

Details

Data files are stored in an operating system appropriate location. Run `pg_cache$cache_path_get()` to get the storage location on your machine. See [pg_cache](#) for more information, including how to set a different base path for downloaded files.

Some files/datasets require the user to be logged in. For now we just pass on these - that is, give back nothing other than metadata.

Value

One or more items of class `pangaea`, each with the `doi`, parent `doi` (if many `dois` within a parent `doi`), `url`, `citation`, `path`, and `data` object. `Data` object depends on what kind of file it is. For tabular data, we print the first 10 columns or so; for a zip file we list the files in the zip (but leave it up to the user to dig unzip and get files from the zip file); for png files, we point the user to read the file in with `png::readPNG()`

Author(s)

Naupaka Zimmerman, Scott Chamberlain

References

<https://www.pangaea.de>

Examples

```
## Not run:
# a single file
(res <- pg_data(doi='10.1594/PANGAEA.807580'))
res[[1]]$doi
res[[1]]$citation
res[[1]]$data
res[[1]]$metadata

# another single file
pg_data(doi='10.1594/PANGAEA.807584')

# Many files
(res <- pg_data(doi='10.1594/PANGAEA.761032'))
res[[1]]
res[[2]]

# Manipulating the cache
## list files in the cache
pg_cache$list()

## clear all data
# pg_cache$delete_all()
pg_cache$list()

## clear a single dataset by DOI
pg_data(doi='10.1594/PANGAEA.812093')
pg_cache$list()
path <- grep("PANGAEA.812093", pg_cache$list(), value = TRUE)
pg_cache$delete(path)
pg_cache$list()

# search for datasets, then pass in DOIs
(searchres <- pg_search(query = 'birds', count = 20))
pg_data(searchres$doi[1])
```

```
# png file
pg_data(doi = "10.1594/PANGAEA.825428")

# zip file
pg_data(doi = "10.1594/PANGAEA.860500")

# login required
## we skip file download
pg_data("10.1594/PANGAEA.788547")

## End(Not run)
```

pg_get_record

Get record from the Pangaea repository

Description

Get record from the Pangaea repository

Usage

```
pg_get_record(identifier, prefix = "oai_dc", as = "df", ...)
```

Arguments

| | |
|------------|--|
| identifier | Dataset identifier. See Examples. |
| 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>oai::get_records()</code> |

Value

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

References

[OAI-PMH documentation](#)

See Also

wraps `oai::get_records()`

Other oai methods: `pg_identify()`, `pg_list_identifiers()`, `pg_list_metadata_formats()`, `pg_list_records()`, `pg_list_sets()`

Examples

```
## Not run:
pg_get_record(identifier = "oai:pangaea.de:doi:10.1594/PANGAEA.788382")
pg_get_record(identifier = "oai:pangaea.de:doi:10.1594/PANGAEA.269656",
  prefix="iso19139")
pg_get_record(identifier = "oai:pangaea.de:doi:10.1594/PANGAEA.269656",
  prefix="dif")

# invalid record id
# pg_get_record(identifier = "oai:pangaea.de:doi:10.1594/PANGAEA.11111")
# pg_get_record(identifier = "oai:pangaea.de:doi:10.1594/PANGAEA.11111",
#   prefix="adfadf")

## End(Not run)
```

pg_identify

Identify information about the Pangaea repository

Description

Identify information about the Pangaea repository

Usage

```
pg_identify(...)
```

Arguments

... [Curl debugging options passed on to `oai::id\(\)`](#)

Value

list

References

[OAI-PMH documentation](#)

See Also

wraps [`oai::id\(\)`](#)

Other oai methods: [`pg_get_record\(\)`](#), [`pg_list_identifiers\(\)`](#), [`pg_list_metadata_formats\(\)`](#), [`pg_list_records\(\)`](#), [`pg_list_sets\(\)`](#)

Examples

```
## Not run:
pg_identify()

## End(Not run)
```

pg_list_identifiers *List identifiers of the Pangaea repository*

Description

List identifiers of the Pangaea repository

Usage

```
pg_list_identifiers(
  prefix = "oai_dc",
  from = NULL,
  until = NULL,
  set = NULL,
  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 timestamp to be used as lower bound for timestamp-based selective harvesting (i.e., only harvest records with timestamps 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 timestamp to be used as an upper bound, for timestamp-based selective harvesting (i.e., only harvest records with timestamps 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>oai::list_identifiers()</code> |

Value

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

References

[OAI-PMH documentation](#)

See Also

wraps `oai::list_identifiers()`

Other oai methods: `pg_get_record()`, `pg_identify()`, `pg_list_metadata_formats()`, `pg_list_records()`, `pg_list_sets()`

Examples

```
## Not run:
pg_list_identifiers(
  from = paste0(Sys.Date() - 4, "T00:00:00Z"),
  until = paste0(Sys.Date() - 3, "T18:00:00Z")
)
pg_list_identifiers(set="geocode1", from=Sys.Date()-1, until=Sys.Date())
pg_list_identifiers(prefix="iso19139", from=Sys.Date()-1, until=Sys.Date())
pg_list_identifiers(prefix="dif",
  from = paste0(Sys.Date() - 2, "T00:00:00Z"),
  until = paste0(Sys.Date() - 1, "T18:00:00Z")
)

## End(Not run)
```

pg_list_metadata_formats

Get metadata formats from the Pangaea repository

Description

Get metadata formats from the Pangaea repository

Usage

```
pg_list_metadata_formats(...)
```

Arguments

... Curl debugging options passed on to `oai::list_metadataformats()`

Value

data.frame

References

[OAI-PMH documentation](#)

See Also

wraps `oai::list_metadataformats()`

Other oai methods: `pg_get_record()`, `pg_identify()`, `pg_list_identifiers()`, `pg_list_records()`, `pg_list_sets()`

Examples

```
## Not run:
pg_list_metadata_formats()

## End(Not run)
```

| | |
|-----------------|----------------------------------|
| pg_list_records | <i>List records from Pangaea</i> |
|-----------------|----------------------------------|

Description

List records from Pangaea

Usage

```
pg_list_records(
  prefix = "oai_dc",
  from = NULL,
  until = NULL,
  set = NULL,
  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>oai::list_records()</code> |

Value

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

References

[OAI-PMH documentation](#)

See Also

wraps `oai::list_records()`

Other oai methods: `pg_get_record()`, `pg_identify()`, `pg_list_identifiers()`, `pg_list_metadata_formats()`, `pg_list_sets()`

Examples

```
## Not run:
pg_list_records(set='citable', from=Sys.Date()-1, until=Sys.Date())

# When no results found > "'noRecordsMatch'"
# pg_list_records(set='geomound', from='2015-01-01', until='2015-01-01')

pg_list_records(prefix="iso19139", set='citable', from=Sys.Date()-1,
  until=Sys.Date())

## FIXME - below are broken
# pg_list_records(prefix="dif", set='citable', from=Sys.Date()-4,
#   until=Sys.Date())
# pg_list_records(prefix="dif", set='project4094', from=Sys.Date()-4,
#   until=Sys.Date())

## End(Not run)
```

pg_list_sets

List the set structure of the Pangaea repository

Description

List the set structure of the Pangaea repository

Usage

```
pg_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>oai::list_sets()</code> |

Value

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

References

[OAI-PMH documentation](#)

See Also

wraps `oai::list_sets()`

Other oai methods: `pg_get_record()`, `pg_identify()`, `pg_list_identifiers()`, `pg_list_metadata_formats()`, `pg_list_records()`

Examples

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

pg_search

Search the Pangea database

Description

Search the Pangea database

Usage

```
pg_search(
  query,
  count = 10,
  offset = 0,
  topic = NULL,
  bbox = NULL,
  mindate = NULL,
  maxdate = NULL,
  ...
)
```

Arguments

| | |
|------------------|--|
| query | (character) Query terms. You can refine a search by prefixing the term(s) with a category, one of citation, reference, parameter, event, project, campaign, or basis. See examples. |
| count | (integer) Number of items to return. Default: 10. Maximum: 500. Use offset parameter to page through results - see examples |
| offset | (integer) Record number to start at. Default: 0 |
| topic | (character) topic area: one of NULL (all areas), "Agriculture", "Atmosphere", "Biological Classification", "Biosphere", "Chemistry", "Cryosphere", "Ecology", "Fisheries", "Geophysics", "Human Dimensions", "Lakes & Rivers", "Land Surface", "Lithosphere", "Oceans", "Paleontology" |
| bbox | (numeric) A bounding box, of the form: minlon, minlat, maxlon, maxlat |
| mindate, maxdate | (character) Dates to search for, of the form "2014-10-28" |
| ... | Curl options passed on to <code>curl::verb-GET</code> |

Details

This is a thin wrapper around the GUI search interface on the page <https://www.pangaea.de>. Everything you can do there, you can do here.

Value

tibble/data.frame with the structure:

- score: match score, higher is a better match
- doi: the DOI for the data package
- size: size number
- size_measure: size measure, one of "data points" or "datasets"
- citation: citation for the data package
- supplement_to: citation for what the data package is a supplement to

See Also

[pg_search_es\(\)](#)

Examples

```
## Not run:
pg_search(query='water')
pg_search(query='water', count=2)
pg_search(query='water', count=20)
pg_search(query='water', mindate="2013-06-01", maxdate="2013-07-01")
pg_search(query='water', bbox=c(-124.2, 41.8, -116.8, 46.1))
pg_search(query='reference:Archer')
pg_search(query='parameter:"carbon dioxide"')
pg_search(query='event:M2-track')
pg_search(query='event:TT011_2-CTD31')
pg_search(query='project:Joint Global Ocean Flux Study')
pg_search(query='campaign:M2')
pg_search(query='basis:Meteor')

# paging with count and offset
# max is 500 records per request - if you need > 500, use offset and count
res1 <- pg_search(query = "florisphaera", count = 500, offset = 0)
res2 <- pg_search(query = "florisphaera", count = 500, offset = 500)
res3 <- pg_search(query = "florisphaera", count = 500, offset = 1000)
do.call("rbind.data.frame", list(res1, res2, res3))

# get attributes: maxScore, totalCount, and offset
res <- pg_search(query='water', bbox=c(-124.2, 41.8, -116.8, 46.1))
attributes(res)
attr(res, "maxScore")
attr(res, "totalCount")
attr(res, "offset")

# curl options
pg_search(query='citation:Archer', verbose = TRUE)

## End(Not run)
```

pg_search_es

Search the Pangaea database with Elasticsearch

Description

Search the Pangaea database with Elasticsearch

Usage

```
pg_search_es(
  query = NULL,
```

```

    size = 10,
    from = NULL,
    source = NULL,
    df = NULL,
    analyzer = NULL,
    default_operator = NULL,
    explain = NULL,
    sort = NULL,
    track_scores = NULL,
    timeout = NULL,
    terminate_after = NULL,
    search_type = NULL,
    lowercase_expanded_terms = NULL,
    analyze_wildcard = NULL,
    version = FALSE,
    ...
)

```

Arguments

| | |
|------------------|--|
| query | (character) Query terms.. |
| size | (character) The number of hits to return. Pass in as a character string to avoid problems with large number conversion to scientific notation. Default: 10. The default maximum is 10,000 - however, you can change this default maximum by changing the <code>index.max_result_window</code> index level parameter. |
| from | (character) The starting from index of the hits to return. Pass in as a character string to avoid problems with large number conversion to scientific notation. Default: 0 |
| source | (character) character vector of fields to return |
| df | (character) The default field to use when no field prefix is defined within the query. |
| analyzer | (character) The analyzer name to be used when analyzing the query string. |
| default_operator | (character) The default operator to be used, can be AND or OR. Default: OR |
| explain | (logical) For each hit, contain an explanation of how scoring of the hits was computed. Default: FALSE |
| sort | (character) Sorting to perform. Can either be in the form of <code>fieldName</code> , or <code>fieldName:asc/fieldName:desc</code> . The <code>fieldName</code> can either be an actual field within the document, or the special <code>_score</code> name to indicate sorting based on scores. There can be several sort parameters (order is important). |
| track_scores | (logical) When sorting, set to TRUE in order to still track scores and return them as part of each hit. |
| timeout | (numeric) A search timeout, bounding the search request to be executed within the specified time value and bail with the hits accumulated up to that point when expired. Default: no timeout. |

| | |
|--------------------------|--|
| terminate_after | (numeric) The maximum number of documents to collect for each shard, upon reaching which the query execution will terminate early. If set, the response will have a boolean field <code>terminated_early</code> to indicate whether the query execution has actually terminated_early. Default: no terminate_after |
| search_type | (character) The type of the search operation to perform. Can be <code>query_then_fetch</code> (default) or <code>dfs_query_then_fetch</code> . Types <code>scan</code> and <code>count</code> are deprecated. |
| lowercase_expanded_terms | (logical) Should terms be automatically lowercased or not. Default: TRUE. |
| analyze_wildcard | (logical) Should wildcard and prefix queries be analyzed or not. Default: FALSE |
| version | (logical) Print the document version with each document. |
| ... | Curl options passed on to curl::verb-GET |

Details

An interface to Pangaea's Elasticsearch query interface. You can also just use `elastic` package to interact with it. The base URL is https://ws.pangaea.de/es/pangaea/panmd/_search

Value

tibble/data.frame, empty if no results

See Also

[pg_search\(\)](#)

Examples

```
## Not run:
(res <- pg_search_es())
attributes(res)
attr(res, "total")
attr(res, "max_score")

pg_search_es(query = 'water', source = c('parentURI', 'minElevation'))
pg_search_es(query = 'water', size = 3)
pg_search_es(query = 'water', size = 3, from = 10)

pg_search_es(query = 'water sky', default_operator = "OR")
pg_search_es(query = 'water sky', default_operator = "AND")

pg_search_es(query = 'water', sort = "minElevation")
pg_search_es(query = 'water', sort = "minElevation:desc")

## End(Not run)
```


Index

* oai methods

- pg_get_record, 6
- pg_identify, 7
- pg_list_identifiers, 8
- pg_list_metadata_formats, 9
- pg_list_records, 10
- pg_list_sets, 11

crul::verb-GET, 4, 13, 16

lapply(), 3

- oai::get_records(), 6
- oai::id(), 7
- oai::list_identifiers(), 8, 9
- oai::list_metadataformats(), 9, 10
- oai::list_records(), 11
- oai::list_sets(), 12

- pangaeear (pangaeear-package), 2
- pangaeear-package, 2
- pg_cache, 2, 2, 4
- pg_cache_clear, 3
- pg_cache_list, 4
- pg_data, 4
- pg_data(), 2
- pg_get_record, 6, 7, 9–12
- pg_identify, 6, 7, 9–12
- pg_list_identifiers, 6, 7, 8, 10–12
- pg_list_metadata_formats, 6, 7, 9, 9, 11, 12
- pg_list_records, 6, 7, 9, 10, 10, 12
- pg_list_sets, 6, 7, 9–11, 11
- pg_search, 12
- pg_search(), 16
- pg_search_es, 14
- pg_search_es(), 14
- png::readPNG(), 5