Package ‘rbace’

October 14, 2022

Type Package

Title 'Bielefeld' Academic Search Engine ('BASE') Client

Description Interface to the API for the 'Bielefeld' Academic Search Engine ('BASE') (<https://www.base-search.net/>). 'BASE' is a search engine for more than 150 million scholarly documents from more than 7000 sources. Methods are provided for searching for documents, as well as getting information on higher level groupings of documents: collections and repositories within collections. Search includes faceting, so you can get a high level overview of number of documents across a given variable (e.g., year). 'BASE' asks users to respect a rate limit, but does not enforce it themselves; we enforce that rate limit.

Version 0.2.2

License MIT + file LICENSE

URL https://docs.ropensci.org/rbace/, https://github.com/ropensci/rbace

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

Encoding UTF-8

Language en-US

LazyLoad yes

Imports crul, xml2, data.table, tibble

Suggests testthat, vcr, webmockr

RoxygenNote 7.1.1

X-schema.org-applicationCategory Literature

X-schema.org-keywords text-ming, literature, scholarly, citations, search

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 2020-10-13 04:20:02 UTC

R topics documented:

`rbace-package` *rbace*

Description

Bielefeld Academic Search Engine Client

Author(s)

Scott Chamberlain

`bs_profile` *Get the profile for a repository*

Description

Get the profile for a repository

Usage

```
bs_profile(target, ...)
```

Arguments

`target`	(character) Internal name of a single repository as delivered in `bs_repositories()`
`...`	curl options passed on to crul::verb-GET

Value

a data.frame, of two columns: "name", "value". "name" holds the "value" description. you can pivot the data.frame to wide by e.g., `tidyr::pivot_wider(x)`

`bs_repositories` *List repositories for a collection*

Description

List repositories for a collection

Usage

```
bs_repositories(coll, ...)
```

Arguments

`coll`	(character) collection code. For existing, pre-defined collections see Appendix, section "Collection-related queries" in the Appendix of https://www.base-search.net/about/download/base
`...`	curl options passed on to crul::verb-GET

Value

a data.frame of two columns: "name", "internal_name"

`bs_retry_options` *bs_search retry options*

Description

bs_search retry options

Usage

```
bs_retry_options(
  pause_base = 1,
  pause_cap = 60,
  pause_min = 1,
  times = 3,
  terminate_on = NULL,
  retry_only_on = NULL,
  onwait = NULL
)
```

Arguments

`pause_base, pause_cap, pause_min`	basis, maximum, and minimum for calculating wait time for retry.
`times`	the maximum number of times to retry.
`terminate_on, retry_only_on`	a vector of HTTP status codes.
`onwait`	a callback function if the request will be retried and a wait time is being applied.

Details

see crul::HttpClient for more detailed explanation of these parameters

Value

a named list with the parameters given to this function

`bs_search` *Search BASE*

Description

Search BASE

Usage

```
bs_search(
  query = NULL,
  target = NULL,
  coll = NULL,
  boost_oa = FALSE,
  hits = NULL,
  offset = NULL,
  fields = NULL,
  sortby = NULL,
  facets = NULL,
  facet_limit = 100,
  facet_sort = NULL,
  filter = NULL,
  raw = FALSE,
  parse = "df",
  retry = bs_retry_options(),
  ...
)

bs_meta(x)
```

Arguments

`query`	(character) query string. For syntax details see Appendix, section "Query syntax"
`target`	(character) Internal name of a single repository as delivered in `bs_repositories()`
`coll`	(character) collection code. For existing, pre-defined collections see Appendix, section "Collection-related queries"
`boost_oa`	(logical) Push open access documents upwards in the result list. Default: `FALSE`
`hits`	(integer) number of results to return. Default: 10. Max: 100
`offset`	(integer) record to start at. Default: 0. Max: 1000
`fields`	(character) Fields to return. This doesn't appear to be working though. The result records only contain fields listed in the comma-separated field list. For existing, pre-defined fields see Appendix, section "Fields"
`sortby`	(character) field to sort by. A sort ordering must include a single field name (see Appendix, section "Fields", table column "Sorting"), followed by a whitespace (escaped as + or %20 in URL strings), followed by sort direction (asc or desc). Default: sorts by relevance
`facets`	(character) The response contains an extra section "facet_counts/facet_fields" with fields from the comma-separated facets list. This section provides a breakdown or summary of the results. From the user's perspective, faceted search breaks up search results into multiple categories, typically showing counts for each, and allows the user to "drill down" or further restrict their search results based on those facets. Use of faceting does not affect the results section of a search response. For existing, pre-defined facet fields see Appendix, section "Fields", table column "Facet".
`facet_limit`	(numeric) Maximum number of constraint counts that should be returned for the facet fields. Default: 100; min:1; max: 500
`facet_sort`	(character) Ordering of the facet field constraints: count - sort by count (highest count first); index - alphabetical sorting. Default: count
`filter`	(character) a string with the value to be used. html escaping will be automatically done; embed string in `I()` to avoid html escaping. This parameter gets used by `fq` solr parameter on the server
`raw`	(logical) If `TRUE` returns raw XML, default: `FALSE`
`parse`	(character) One of 'list' or 'df'
`retry`	(list) use `bs_retry_options()` to make a named list of retry options to pass on to the HTTP request. default values are passed for you, but you can change them by setting an option in `bs_retry_options()`
`...`	Facet field based query options (See Facet below) or curl options passed on to crul::verb-GET
`x`	input to `bs_meta`

Details

BASE asks that requests are not more frequent than 1 per second, so we enforce the rate limit internally. if you do a single request not in a for loop/lapply type situation, this won't be inoked, but will if doing a for loop/lapply call, and there's no sleep invoked

Value

XML as character string if `parse = FALSE` or data.frame

Facet

You can optionally pass in search term for specific facet fields. See example. For existing, predefined facet fields see Appendix at https://www.base-search.net/about/download/base_interface.pdf, section "Fields", table column "Facet"

Index