Package 'rscopus'

May 9, 2026

Type Package

Title Scopus Database 'API' Interface

Version 0.9.0

Maintainer John Muschelli <muschellij2@gmail.com>

Description Uses Elsevier 'Scopus' API <https://dev.elsevier.com/sc_apis.html> to download information about authors and their citations.

License GPL-2

Depends R (>= 3.0.0)

Imports httr, jsonlite, utils, stats, plyr, tidyr, dplyr, tools, glue, magrittr

Suggests xml2, rvest, graphics, testthat, jpeg, knitr, rmarkdown, purrr, pbapply

URL https://dev.elsevier.com/sc_apis.html, https://github.com/muschellij2/rscopus

BugReports https://github.com/muschellij2/rscopus/issues

RoxygenNote 7.3.2

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Author John Muschelli [aut, cre]

Repository CRAN

Date/Publication 2025-07-30 23:40:02 UTC

Contents

`abstract_retrieval` *SCOPUS Abstract Retrieval*

Description

This function wraps `generic_elsevier_api` to give a retrieval of an abstract from the Elsevier Abstract Retrieval API

Usage

```
abstract_retrieval(
  id,
 identifier = c("scopus_id", "eid", "doi", "pii", "pubmed_id", "pui", "group_id"),
  http_end = NULL,
  ...
)
```

Arguments

`id`	Identifier for abstract
`identifier`	Type of identifier to use
`http_end`	any additional end to http statement. See `generic_elsevier_api`
`...`	Arguments to be passed to `generic_elsevier_api`

Value

List of elements, similar to `generic_elsevier_api`

See Also

`generic_elsevier_api`

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = abstract_retrieval("S1053811915002700", identifier = "pii",
   verbose = FALSE)
   x = abstract_retrieval("S1053811915002700", identifier = "pii",
   view = "FULL",
   verbose = FALSE)
}
```

`affiliation_retrieval` *SCOPUS Affiliation Retrieval*

Description

This function wraps `generic_elsevier_api` to give a retrieval of an affiliation from the Elsevier Affiliation Retrieval API

Usage

```
affiliation_retrieval(
  id,
  identifier = c("affiliation_id", "eid"),
  http_end = NULL,
  ...
)
```

Arguments

`id`	Identifier for affiliation
`identifier`	Type of identifier to use
`http_end`	any additional end to http statement. See `generic_elsevier_api`
`...`	Arguments to be passed to `generic_elsevier_api`

Value

List of elements, similar to `generic_elsevier_api`

See Also

`generic_elsevier_api`

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = affiliation_retrieval("60006183", identifier = "affiliation_id",
   verbose = FALSE)
}
```

affil_df *Search Affiliation Content on SCOPUS*

Description

Searches SCOPUS to get information about documents on an affiliation

Usage

```
affil_df(
  affil_id = NULL,
  affil_name = NULL,
  api_key = NULL,
  verbose = TRUE,
  facets = NULL,
  sort = "document-count",
  ...
)

affil_data(
  affil_id = NULL,
  affil_name = NULL,
  api_key = NULL,
  verbose = TRUE,
  facets = NULL,
  sort = "document-count",
  ...
)
```

Arguments

affil_id	Affiliation ID number.
affil_name	affiliation name. Disregarded if affil_id is specified
api_key	Elsevier API key
verbose	Print diagnostic messages
facets	Facets sent in query. See https://dev.elsevier.com/api_docs.html
sort	sorting sent to query
...	Arguments to be passed to author_search

Value

List of entries from SCOPUS

Note

The affil_data command will return the list of all entries as well as the data.frame

See Also

get_author_info

---

affil_list_to_df *List of SCOPUS Entries to List of Affiliations Data Frames*

Description

Take a SCOPUS entry and transform it to a data frame of affiliations

Usage

```
affil_list_to_df(affils)
```

Arguments

affils	List of affiliations, from entries_to_affil_list

Value

A data.frame of affiliation information. A column named "index" denotes the element of the object affils that the row corresponds to

---

affil_search *Search Author Content on SCOPUS*

Description

Searches SCOPUS to get information about documents on an author.

Usage

```
affil_search(
  affil_id,
  searcher = "AF-ID",
  http = "https://api.elsevier.com/content/search/affiliation",
  facets = "affilcountry(sort=document-count)",
  count = 200,
  ...
)
```

Arguments

affil_id	Affiliation ID number
searcher	Identifier for Affiliation ID. Do not change unless you know exactly what the API calls for.
http	Address for scopus api
facets	Facets sent in query. See https://dev.elsevier.com/api_docs.html
count	number of records to retrieve (below 200, see https://dev.elsevier.com/api_key_settings.html)
...	Arguments to be passed to author_search

Value

List of entries from SCOPUS

See Also

get_author_info

---

all_possible_affils *Find all affiliations*

Description

Take a SCOPUS entry and transform it to a data frame of all affiliations listed in there

Usage

```
all_possible_affils(entries)
```

Arguments

entries	list of entries from SCOPUS, usually from author_search result

Value

A data.frame of affiliations

article_retrieval *ScienceDirect Article Retrieval*

Description

This function wraps generic_elsevier_api to give a retrieval of an article from the Elsevier Article Retrieval API

Usage

```
article_retrieval(
  id,
  view = c("META", "META_ABS", "META_ABS_REF", "FULL", "REF", "ENTITLED"),
  identifier = c("scopus_id", "eid", "doi", "pii", "pubmed_id"),
  http_end = NULL,
  ...
)
```

Arguments

id	Identifier for article
view	Which view to see. See https://dev.elsevier.com/guides/ArticleRetrievalViews.htm
identifier	Type of identifier to use
http_end	any additional end to http statement. See generic_elsevier_api
...	Arguments to be passed to generic_elsevier_api

Value

List of elements, similar to generic_elsevier_api

See Also

generic_elsevier_api

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
    x = article_retrieval("S1053811915002700", identifier = "pii",
    verbose = FALSE, view = "FULL")
    gen = x$content$`full-text-retrieval-response`
    ot = gen$originalText
} else {
 x = article_retrieval("S1053811915002700",
   identifier = "pii",
   api_key_error = FALSE)
}
```

author_df *Search Author Content on SCOPUS*

Description

Searches SCOPUS to get information about documents on an author. Note, `author_list` returns a list of the entries from `author_search`, but allows you to put in a name.

Usage

```
author_df(
  au_id = NULL,
  last_name = NULL,
  first_name = NULL,
  api_key = NULL,
  verbose = TRUE,
  all_author_info = FALSE,
  http = "https://api.elsevier.com/content/search/scopus",
  view = "COMPLETE",
  count = 25,
  general = TRUE,
  scrub = FALSE,
  headers = NULL,
  ...
)

author_df_orig(..., general = FALSE)

author_list(
  au_id = NULL,
  last_name = NULL,
  first_name = NULL,
  api_key = NULL,
  verbose = TRUE,
  http = "https://api.elsevier.com/content/search/scopus",
  view = "COMPLETE",
  count = 25,
  headers = NULL,
  ...
)

author_data(
  ...,
  verbose = TRUE,
  all_author_info = FALSE,
  general = TRUE,
  scrub = FALSE
```

)

Arguments

au_id	Author ID number. Overrides any first/last name argument
last_name	last name of author
first_name	first name of author
api_key	Elsevier API key
verbose	Print diagnostic messages
all_author_info	Should all author info be recorded instead of that just to the author given
http	Address for scopus api
view	type of view to give, see https://dev.elsevier.com/documentation/ScopusSearchAPI.wadl
count	number of records to retrieve (below 25, see https://dev.elsevier.com/api_key_settings.html)
general	Should gen_entries_to_df instead of the way before version 0.5.10.9001
scrub	Should 'scrub_identifier' be run on the identifier?
headers	Headers passed to add_headers, passed to GET
...	Arguments to be passed to author_search

Value

List of entries from SCOPUS

Note

The author_data command will return the list of all entries as well as the data.frame.

See Also

get_author_info

Examples

```
if (is_elsevier_authorized()) {
res = author_df(last_name = "Muschelli", first_name = "John",
verbose = FALSE)
}
```

`author_retrieval_id` *SCOPUS Author Retrieval*

Description

This function wraps `generic_elsevier_api` to give a retrieval of an author from the Elsevier Author Retrieval API

Usage

```
author_retrieval_id(
  id,
  identifier = c("author_id", "eid"),
  http_end = NULL,
  ...
)

multi_author_retrieval(
  id,
  identifier = c("author_id", "eid"),
  view = c("LIGHT", "STANDARD", "ENHANCED", "METRICS", "ENTITLED"),
  http_end = NULL,
  ...
)

author_retrieval(
  au_id = NULL,
  last_name = NULL,
  first_name = NULL,
  view = c("LIGHT", "STANDARD", "ENHANCED", "METRICS", "ENTITLED"),
  self_cite = c("include", "exclude"),
  http_end = NULL,
  verbose = TRUE,
  api_key = NULL,
  headers = NULL,
  ...
)
```

Arguments

`id`	Identifier for author
`identifier`	Type of identifier to use
`http_end`	any additional end to http statement. See `generic_elsevier_api`
`...`	Arguments to be passed to `generic_elsevier_api`
`view`	Which view to see. See https://dev.elsevier.com/documentation/AuthorRetrievalAPI.wadl

au_id	Author ID number. Overrides any first/last name argument
last_name	last name of author
first_name	first name of author
self_cite	Should self-citations be included?
verbose	Print diagnostic messages
api_key	Elsevier API key
headers	Headers passed to add_headers, passed to GET

Value

List of elements, similar to generic_elsevier_api

Note

See https://dev.elsevier.com/documentation/AuthorRetrievalAPI.wadl for documentation

See Also

generic_elsevier_api

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = author_retrieval(au_id = "40462056100",
   verbose = FALSE)
   x = author_retrieval_id("40462056100", identifier = "author_id",
   verbose = FALSE)
} else {
 x = author_retrieval_id(
 "40462056100",
 identifier = "author_id",
   api_key_error = FALSE, verbose = FALSE)
 x = author_retrieval(
 au_id = "40462056100",
   api_key_error = FALSE, verbose = FALSE)
}
```

---

author_search *Search Author Content on SCOPUS*

Description

Searches SCOPUS to get information about documents on an author.

Usage

```
author_search(
  au_id,
  api_key = NULL,
  http = "https://api.elsevier.com/content/search/author",
  count = 200,
  start = 0,
  verbose = TRUE,
  facets = "subjarea(sort=fd,count=350)",
  searcher = "AU-ID",
  max_count = Inf,
  view = c("STANDARD", "COMPLETE"),
  add_query = NULL,
  headers = NULL,
  wait_time = 0,
  ...
)
```

Arguments

au_id	Author ID number
api_key	API Key for Elsevier
http	Address for scopus api
count	number of records to retrieve (below 200, see https://dev.elsevier.com/api_key_settings.html)
start	where should the records start gathering
verbose	Print diagnostic messages
facets	Facets sent in query. See https://dev.elsevier.com/api_docs.html
searcher	Identifier for author ID. Do not change unless you know exactly what the API calls for.
max_count	Maximum count of records to be returned.
view	type of view to give, see https://dev.elsevier.com/documentation/AuthorSearchAPI.wadl
add_query	Things to add to the query parameter for the request
headers	additional headers to be added to add_headers
wait_time	The time in seconds to wait across consecutive requests of a single search (when records > 25)
...	Arguments to be passed to the query list for GET

Value

List of entries from SCOPUS

See Also

get_author_info

Examples

```
## Not run:
author_search(au_id = "Smith", searcher = "affil(princeton) and authlast")
berk = author_search(au_id = "berkeley", searcher = "affil", count =100)

## End(Not run)
```

`author_search_by_affil` *Search Authors by Affiliation on SCOPUS*

Description

Searches SCOPUS to get information about authors with a certain affiliation

Usage

```
author_search_by_affil(affil_id, searcher = "AF-ID", ...)
```

Arguments

`affil_id`	Affiliation ID number
`searcher`	Identifier for Affiliation ID. Do not change unless you know exactly what the API calls for.
`...`	Arguments to be passed to `GET`

Value

List of entries from SCOPUS

See Also

`get_author_info`

`bibtex_core_data` *Makes a* `bibtex` *entry from an output of* `abstract_retrieval` *or* `article_retrieval`

Description

Makes a `bibtex` entry from an output of `abstract_retrieval` or `article_retrieval`

Usage

```
bibtex_core_data(x)
```

Arguments

x	output of abstract_retrieval or article_retrieval, with both get_statement and content

Value

A character vector of bibtex values

Note

Adapted from https://github.com/pybliometrics-dev/pybliometrics/blob/master/pybliometrics/scopus/abstract_retrieval.py

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = abstract_retrieval("S1053811915002700", identifier = "pii",
   verbose = FALSE)
   res = bibtex_core_data(x)
   cat(res)
   x = abstract_retrieval("84929707579", identifier = "scopus_id",
   verbose = FALSE)
   res2 = bibtex_core_data(x)
   cat(res2)
}
```

---

citation_retrieval *SCOPUS Citation Retrieval*

Description

SCOPUS Citation Retrieval

Usage

```
citation_retrieval(
  scopus_id = NULL,
  pii = NULL,
  doi = NULL,
  pubmed_id = NULL,
  date_range = NULL,
  exclude = NULL,
  ...
)

parse_citation_retrieval(result)
```

Arguments

scopus_id	Scopus Identifier
pii	Scopus Identifier
doi	Scopus Identifier
pubmed_id	PubMed Identifier
date_range	date range to specify, must be length 2
exclude	either exclude-self or exclude-books for exclusion of citation
...	Arguments to be passed to generic_elsevier_api
result	result from citation_retrieval, which has an element of content

Value

List of elements, similar to generic_elsevier_api

See Also

generic_elsevier_api

Examples

```
api_key = Sys.getenv("Elsevier_API_Interactive")
set_api_key(api_key)
if (!is.null(api_key) & nchar(api_key) > 0){
  result = citation_retrieval(pii = c("S0140673616324102",
                                      "S0014579301033130"),
                              verbose = FALSE)
  if (httr::status_code(result$get_statement) < 400) {
    res = parse_citation_retrieval(result)
  }

}
set_api_key(NULL)
```

---

collapse_affil *Collapse Affiliations to one row*

Description

Take an individual SCOPUS entry and transform it to a data frame of affiliations

Usage

```
collapse_affil(affils, collapse = ";")
```

Arguments

affils	Data frame of affiliations to be collapsed usually from author_search
collapse	What should values be collapsed using as a separator

Value

A data.frame of affiliation information

---

complete_multi_author_info *Get Complete Author Information and ID from Scopus*

Description

Get Complete Author Information and ID from Scopus

Usage

```
complete_multi_author_info(au_id = NULL, api_key = NULL, verbose = TRUE, ...)
```

Arguments

au_id	vector of Author IDs
api_key	Elsevier API key
verbose	Print messages from specification
...	options to pass to generic_elsevier_api

Value

List of information

---

elsevier_authenticate *Authenticate API Key and get Token*

Description

Authenticate API Key and get Token

Usage

```
elsevier_authenticate(
  api_key = NULL,
  api_key_error = TRUE,
  choice = NULL,
  verbose = TRUE,
  headers = NULL,
  ...
)

is_elsevier_guest(...)

is_elsevier_authorized(...)
```

Arguments

api_key	Elsevier API key
api_key_error	Should there be an error if no API key?
choice	Choice of which registered See https://dev.elsevier.com/tecdoc_api_authentication.html
verbose	Print messages from specification
headers	Headers passed to add_headers, passed to GET
...	Additional arguments to send to GET.

Value

List of content, the GET request, and the token

Examples

```
if (have_api_key()) {
   auth = elsevier_authenticate()
}
```

---

embase_retrieval *SCOPUS Embase Retrieval*

Description

This function wraps generic_elsevier_api to give a retrieval of an Embase Entry from the Elsevier Embase Retrieval API

Usage

```
embase_retrieval(
  id,
  identifier = c("lui", "pii", "doi", "embase", "pubmed_id", "medline"),
  http_end = NULL,
  ...
)
```

Arguments

id	Identifier for abstract
identifier	Type of identifier to use
http_end	any additional end to http statement. See generic_elsevier_api
...	Arguments to be passed to generic_elsevier_api

Value

List of elements, similar to generic_elsevier_api

See Also

generic_elsevier_api

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = embase_retrieval("S1053811915002700", identifier = "pii",
   verbose = FALSE)
}
```

---

entitlement_retrieval *ScienceDirect Text Entitlement Retrieval*

Description

This function wraps generic_elsevier_api to give a retrieval of an entitlement from the Elsevier Text Entitlement API

Usage

```
entitlement_retrieval(
  id,
  identifier = c("scopus_id", "eid", "doi", "pii", "pubmed_id"),
  http_end = NULL,
  ...
)
```

Arguments

id	Identifier for entitlement
identifier	Type of identifier to use
http_end	any additional end to http statement. See generic_elsevier_api
...	Arguments to be passed to generic_elsevier_api

Value

List of elements, similar to generic_elsevier_api

See Also

generic_elsevier_api

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = entitlement_retrieval("S1053811915002700", identifier = "pii",
   verbose = FALSE)
}
```

---

entries_to_affil_list *List of SCOPUS Entries to List of Affiliations Data Frames*

Description

Take a SCOPUS entry and transform it to a data frame of affiliations

Usage

```
entries_to_affil_list(entries)
```

Arguments

entries	list of entries from SCOPUS, usually from author_search result

Value

A data.frame of affiliation information

entries_to_citation_df *List of SCOPUS Entries to Data Frame of Citations*

Description

Take a SCOPUS entry list and transform it to a data frame of citation and article information

Usage

```
entries_to_citation_df(entries)
```

Arguments

entries	list of entries from SCOPUS, usually from author_search result

Value

A data.frame of citation information

entries_to_df *Convert Entries into a data.frame*

Description

Converts a list of entries into a data.frame of records

Usage

```
entries_to_df(entries, au_id = NULL, verbose = TRUE)

entries_to_df2(entries, verbose = TRUE)
```

Arguments

entries	Entries from the output of a command
au_id	Author ID number. Overrides any first/last name argument
verbose	Print diagnostic messages

Value

Data frame of records

entry_to_affil *SCOPUS Entry to Affiliation*

Description

Take an individual SCOPUS entry and transform it to a data frame of affiliations

Usage

```
entry_to_affil(x, all_affils)
```

Arguments

x	individual entry from SCOPUS, usually from author_search
all_affils	Affiliation data.frame from all_possible_affils

Value

A data.frame of affiliation information

generic_elsevier_api *Generic Elsevier Search*

Description

Runs GET on generic Elsevier Search

Usage

```
generic_elsevier_api(
  query = NULL,
  type = c("search", "article", "entitlement", "recommendation", "object", "fragment",
   "abstract", "affiliation", "embase", "author", "serial", "nonserial", "subject",
    "holdings", "citation-count", "citations", "metadata", "ev", "ev_records",
    "analytics"),
  search_type = c("affiliation", "author", "scopus", "scidir", "scidir-object",
    "sciencedirect", "plumx"),
  api_key = NULL,
  headers = NULL,
  content_type = c("content", "feedback"),
  root_http = "https://api.elsevier.com",
  http_end = NULL,
  verbose = TRUE,
  api_key_error = TRUE,
  ...
)
```

Arguments

query	Query to run, not overall query, but 'queryParam' query
type	Type of search. See https://dev.elsevier.com/api_docs.html
search_type	Type of search if type = "search". See https://dev.elsevier.com/api_docs.html
api_key	Elsevier API key
headers	Headers passed to add_headers, passed to GET
content_type	Is the data content or feedback?
root_http	address to use for query
http_end	string to add to end of http specification (done using paste0)
verbose	Print messages from specification
api_key_error	Should there be an error if no API key?
...	Options passed to queryParam for GET

Value

List of elements, content and the GET request

Examples

```
## Not run:
query_string = "affil(hopkins)"
# Use affiliation query
s = generic_elsevier_api(query = query_string,
                         type = "search", search_type = "affiliation",
                         api_key = api_key,
                       verbose = FALSE)

# Use author query
s = generic_elsevier_api(query = query_string,
type = "search", search_type = "author",
api_key = api_key,
                       verbose = FALSE)

# Query abstract by pii
s = generic_elsevier_api(query = "",
                       type = "abstract", http_end = "pii/S1053811915002700",
                       api_key = api_key,
                       verbose = FALSE)

## End(Not run)
```

gen_entries_to_df *Generally Convert Entries into a list of* data.frames

Description

Generally Convert Entries into a list of data.frames

Usage

```
gen_entries_to_df(entries, scrub = FALSE)
```

Arguments

entries	Entries from the output of a command
scrub	Should ‘scrub_identifier‘ be run on the identifier?

Value

List of data.frames from entries

get_affiliation_info *Get Affiliation Information and ID from Scopus*

Description

Uses SCOPUS affiliation search to identify affiliation identification information

Usage

```
get_affiliation_info(
  affil_id = NULL,
  affil_name = NULL,
  api_key = NULL,
  verbose = FALSE,
  headers = NULL
)
```

Arguments

affil_id	ID of affiliation
affil_name	name of affiliation
api_key	Elsevier API key
verbose	Print messages from specification
headers	Headers passed to add_headers, passed to GET

Value

A data.frame of affiliation information

---

get_all_coauthors *Get all Co-Authors*

Description

Get all Co-Authors

Usage

```
get_all_coauthors(...)
```

Arguments

...	arguments to pass to author_df

Value

A list of output, including a vector of author_ids

Examples

```
 ## Not run:
 get_all_coauthors(last_name = "muschelli")

## End(Not run)
```

---

get_api_key *Find API Key for Elsevier*

Description

Determines if option(elsevier_api_key) or option(elsevier_api_key_filename) is set. If not, it stops and returns an error. If so, returns the value.

Usage

```
get_api_key(api_key = NULL, error = TRUE)

have_api_key(api_key = NULL)
```

Arguments

api_key	Elsevier API key
error	Should the function error if api_key = NULL?

Value

API key

Note

You can either set the API key using `option(elsevier_api_key)` or have it accessible by `api_key = Sys.getenv('Elsevier_API')`.

Examples

```
res = get_api_key(error = FALSE)
```

`get_author_info` *Get Relevant Author Information and ID from Scopus in Data Frame*

Description

Uses SCOPUS author search to identify author identification information in a workable format

Usage

```
get_author_info(...)
```

Arguments

`...`	Arguments passed to `get_complete_author_info`

Value

Data.frame of information

See Also

`get_complete_author_info`

Examples

```
## Not run:
get_author_info(au_id = "40462056100")

## End(Not run)
```

get_complete_author_info *Get Complete Author Information and ID from Scopus*

Description

Uses SCOPUS author search to identify author identification information

Usage

```
get_complete_author_info(
  last_name = NULL,
  first_name = NULL,
  affil_id = NULL,
  affil_name = NULL,
  api_key = NULL,
  http = "https://api.elsevier.com/content/search/author",
  query = NULL,
  count = 200,
  start = 0,
  verbose = TRUE,
  au_id = NULL,
  headers = NULL,
  ...
)
```

Arguments

last_name	last name of author
first_name	first name of author
affil_id	ID of affiliation (optional)
affil_name	name of affiliation
api_key	Elsevier API key
http	Author API http
query	Additional query info, added using +AND+ to original query
count	maximum number of records to retrieve
start	index to start on. Only necessary if a large number of records retrieved
verbose	Print messages from specification
au_id	Author ID number, will override first/last combination if specified
headers	Headers passed to add_headers, passed to GET
...	options to pass to GET

Value

List of information

Examples

```
if (is_elsevier_authorized()) {
res = get_complete_author_info(
last_name = "Muschelli",
first_name = "John",
verbose = FALSE)
}
```

`get_links` *Get Links for next/first/last query*

Description

Get Links for next/first/last query

Usage

```
get_links(result)

get_url(result, type = c("self", "first", "prev", "next", "last"))

next_url(result)

last_url(result)

prev_url(result)

self_url(result)

first_url(result)

get_link_type(result, ..., type = c("self", "first", "prev", "next", "last"))

get_first(result, ...)

get_last(result, ...)

get_prev(result, ...)

get_next(result, ...)

get_self(result, ...)
```

Arguments

result	Object (list) with an element named content, usually from generic_elsevier_api
type	The type of link requested
...	Options passed to GET

Value

A data.frame or a vector of characters

Examples

```
## Not run:
result <- generic_elsevier_api(
query = "ISSN(0004-3702) AND YEAR(2001)",
search_type = "scopus")
next_result = get_next(result)


## End(Not run)
```

---

inst_token_header *Add Institution or Authorization Token*

Description

Add Institution or Authorization Token

Usage

```
inst_token_header(token)

auth_token_header(token)
```

Arguments

token	Elsevier API token, usually from elsevier_authenticate

Value

An object of class token, but really a character

metadata_retrieval *SCOPUS Citation Retrieval*

Description

SCOPUS Citation Retrieval

Usage

```
metadata_retrieval(query = NULL, view = c("STANDARD", "COMPLETE"), ...)
```

Arguments

query	Query to run, not overall query, but 'queryParam' query
view	Type of view to have. See https://dev.elsevier.com/guides/ArticleMetadataViews.htm
...	Arguments to be passed to generic_elsevier_api

Value

List of elements, similar to generic_elsevier_api

See Also

generic_elsevier_api, https://dev.elsevier.com/documentation/ArticleMetadataAPI.wadl

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = metadata_retrieval(query = "heart attack",
   verbose = FALSE)
}
```

multi_author_info *Get Relevant Authors Information from IDs from Scopus*

Description

Get Relevant Authors Information from IDs from Scopus

Usage

```
multi_author_info(...)

process_complete_multi_author_info(res)
```

Arguments

...	Arguments passed to get_complete_author_info
res	result from complete_multi_author_info

Value

Data.frame of information

See Also

get_complete_author_info

Examples

```
## Not run:
multi_author_info(au_id =  c("22968535800", "40462056100"))

## End(Not run)
```

---

nonull *Remove NULL*

Description

Removes NULL values from a vector from a list

Usage

```
nonull(x, replace = NA)
```

Arguments

x	Vector from a list
replace	Value to replace NULL with

Value

Vector

`object_retrieval` *ScienceDirect Object Retrieval*

Description

This function wraps `generic_elsevier_api` to give a retrieval of an object from the Elsevier Object Retrieval API

Usage

```
object_retrieval(
  id,
  identifier = c("scopus_id", "eid", "doi", "pii", "pubmed_id"),
  ref = NULL,
  http_end = NULL,
  ...
)

process_object_retrieval(res)

download_object(
  url,
  api_key = NULL,
  api_key_error = TRUE,
  verbose = TRUE,
  headers = NULL,
  ...
)

download_objects(url, ...)
```

Arguments

`id`	Identifier for object
`identifier`	Type of identifier to use
`ref`	document reference
`http_end`	any additional end to http statement. See `generic_elsevier_api`
`...`	Arguments to be passed to `generic_elsevier_api` or `GET`
`res`	result from `object_retrieval`
`url`	url to download from `object_retrieval`
`api_key`	Elsevier API key
`api_key_error`	Should there be an error if no API key?
`verbose`	Print messages from specification
`headers`	Headers passed to `add_headers`, passed to `GET`

Value

List of elements, similar to generic_elsevier_api

See Also

generic_elsevier_api

Examples

```
api_key = get_api_key(NULL, error = FALSE)
authorized = is_elsevier_authorized()
if (have_api_key()){
   x = object_retrieval("S1053811915002700", identifier = "pii",
   verbose = FALSE)
   df = process_object_retrieval(x)
   df = df[ grepl("image/jpeg", df$mime_type),,drop = FALSE ]
   df = df[ df$type %in% "IMAGE-HIGH-RES",,drop = FALSE ]
}
if (authorized) {
   res = download_object(df$url[1])
   if (interactive()) {
      browseURL(res$outfile)
   } else {
     img = res$content
     dims = dim(img)[1:2]
     mdim = max(dims)
     graphics::plot(c(0, ncol(img)), c(0, nrow(img)), type='n')
     graphics::rasterImage(img, 1, 1, ncol(img), nrow(img))
   }
}
```

---

plumx_metrics *Retrieve PlumX metrics for Scopus documents and other related artifacts*

Description

Retrieve PlumX metrics for Scopus documents and other related artifacts

Usage

```
plumx_metrics(value, type = plumx_types(), ...)

plumx_types()
```

Arguments

value	The value of the identifier to search for.
type	The type of identifier to search for.
...	Additional arguments to pass to generic_elsevier_api, other than http_end, type, search_type, and content_type

Value

List of elements, content and the GET request

Note

See https://dev.elsevier.com/documentation/PlumXMetricsAPI.wadl for more information

Examples

```
if (have_api_key()) {
type = "doi"
value = "10.1016/j.nicl.2018.10.013"
res = plumx_metrics(value, type)
}
```

---

print.scopus_api_key *Print method for Scopus API key*

Description

Print method for Scopus API key

Usage

```
## S3 method for class 'scopus_api_key'
print(x, reveal = FALSE, ...)
```

Arguments

x	an object used to select a method.
reveal	Should the API key be revealed
...	further arguments passed to or from other methods

Examples

```
x = "asdf"
class(x) = "scopus_api_key"
print(x)
print(x, reveal = TRUE)
```

print.token *Print method for token*

Description

Print method for token

Usage

```
## S3 method for class 'token'
print(x, reveal = FALSE, ...)

reveal(x, ...)
```

Arguments

x	an object used to select a method.
reveal	Should the token be revealed
...	further arguments passed to or from other methods

Examples

```
x = "asdf"
class(x) = "token"
print(x)
print(x, reveal = TRUE)
```

process_affiliation_name *Process Affiliation Name*

Description

Process Affiliation ID and names for generic use

Usage

```
process_affiliation_name(
  affil_id = NULL,
  affil_name = NULL,
  api_key = NULL,
  verbose = TRUE
)
```

Arguments

affil_id	Affiliation ID number.
affil_name	affiliation name
api_key	Elsevier API key
verbose	Print diagnostic messages

Value

List of affiliation name and affiliation ID

---

process_author_name *Process Author Name*

Description

Process author ID and names for generic use

Usage

```
process_author_name(
  au_id = NULL,
  last_name = NULL,
  first_name = NULL,
  affil_id = NULL,
  api_key = NULL,
  verbose = TRUE,
  headers = NULL
)
```

Arguments

au_id	Author ID number. Overrides any first/last name argument
last_name	last name of author
first_name	first name of author
affil_id	ID of affiliation (optional)
api_key	Elsevier API key
verbose	Print diagnostic messages
headers	Headers passed to add_headers, passed to GET

Value

List of first/last name and author ID

Note

This function is really to avoid duplication

read_cto *Read Citation Overview (CTO) File*

Description

Read Citation Overview (CTO) File

Usage

```
read_cto(file)

read_cto_long(file)
```

Arguments

file	CSV of CTO export from Scopus

Value

A list of the data, year columns, and header information

Examples

```
file = system.file("extdata", "CTOExport.csv", package = "rscopus")
citations_over_time = read_cto(file)
citations_over_time = citations_over_time$data
```

recommendation_retrieval *ScienceDirect Article Recommendation Retrieval*

Description

This function wraps generic_elsevier_api to give a retrieval of a recommendation from the Elsevier Article Recommendation API

Usage

```
recommendation_retrieval(
  id,
  identifier = c("eid", "pii"),
  http_end = NULL,
  ...
)
```

Arguments

id	Identifier for recommendation
identifier	Type of identifier to use
http_end	any additional end to http statement. See generic_elsevier_api
...	Arguments to be passed to generic_elsevier_api

Value

List of elements, similar to generic_elsevier_api

See Also

generic_elsevier_api

Examples

```
api_key = get_api_key(NULL, error = FALSE)
if (!is.null(api_key)){
   x = recommendation_retrieval("S1053811915002700", identifier = "pii",
   verbose = FALSE)
}
```

---

replace_non_ascii *Replace non-ASCII characters*

Description

Replaces non-ASCII characters from a last or first name

Usage

```
replace_non_ascii(string)
```

Arguments

string	Character vector of values to be replaced

Value

Character vector

See Also

Taken from https://stackoverflow.com/questions/20495598/replace-accented-characters-in-r-with-non-ac

scopus_search *SCOPUS Search*

Description

This function wraps generic_elsevier_api to give a scopus search from the Elsevier Scopus Search API

Searches SCOPUS to get information about documents on an author.

Usage

```
scopus_search(
  query,
  api_key = NULL,
  count = 200,
  view = c("STANDARD", "COMPLETE"),
  start = 0,
  verbose = TRUE,
  max_count = 20000,
  http = "https://api.elsevier.com/content/search/scopus",
  headers = NULL,
  wait_time = 0,
  ...
)

sciencedirect_search(count = 100, ...)

scidir_search(count = 100, ...)
```

Arguments

query	Query string to search on SCOPUS
api_key	API Key for Elsevier
count	number of records to retrieve (below 200 for STANDARD, below 25 for COMPLETE views, see https://dev.elsevier.com/api_key_settings.html).
view	type of view to give, see https://dev.elsevier.com/documentation/ScopusSearchAPI.wadl
start	where should the records start gathering
verbose	Print diagnostic messages
max_count	Maximum count of records to be returned.
http	Address for scopus API
headers	additional headers to be added to add_headers
wait_time	The time in seconds to wait across consecutive requests of a single search (when records > 25)
...	Arguments to be passed to the query list for GET

Value

List of entries from SCOPUS

Examples

```
if (have_api_key()) {
  authorized = is_elsevier_authorized()
  if (authorized) {
    res = scopus_search(query = "all(gene)", max_count = 20,
                        count = 10)
    df = gen_entries_to_df(res$entries)
    head(df$df)
    sci_res = sciencedirect_search(query = "heart+attack AND text(liver)",
                                   max_count = 30, count = 25)
    sci_df = gen_entries_to_df(sci_res$entries)
    Sys.sleep(2)
    nt = sciencedirect_search(query = "title(neurotoxin)", max_count = 20,
                              count = 10, wait_time = 1)
    nt_df = gen_entries_to_df(nt$entries)
    nt_df = nt_df$df
  }
}
```

---

set_api_key *Set API Key for Elsevier*

Description

Sets Elsevier API key using if `option(elsevier_api_key)`

Usage

```
set_api_key(api_key)
```

Arguments

`api_key`	Elsevier API key

`subject_areas` *Subject Areas*

Description

Subject Areas

Usage

```
subject_areas()

subject_area_codes()
```

Value

Character vector of subject areas

Note

See https://service.elsevier.com/app/answers/detail/a_id/15181/supporthub/scopus/related/1/

Index