Package ‘openalexR’

October 14, 2022

Type Package

Title Getting Bibliographic Records from 'OpenAlex' Database Using 'DSL' API

Version 1.0.0

Description A set of tools to extract bibliographic content from 'OpenAlex' database using API <https://docs.openalex.org/api/>.

License MIT + file LICENSE

URL https://github.com/massimoaria/openalexR,
https://massimoaria.github.io/openalexR/

BugReports https://github.com/massimoaria/openalexR/issues

Imports curl, httr, jsonlite, progress, tibble

Suggests testthat (>= 3.0.0), dplyr, knitr, rmarkdown, tidyr, purrr, ggplot2

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Config/testthat/edition 3

NeedsCompilation no

Author Massimo Aria [aut, cre, cph] (<https://orcid.org/0000-0002-8517-9411>), Trang Le [aut] (<https://orcid.org/0000-0003-3737-6565>)

Maintainer Massimo Aria <massimo.aria@gmail.com>

Repository CRAN

Date/Publication 2022-10-06 10:40:02 UTC

R topics documented:

`countrycode` *Index of Countries and their alpha-2 and aplha-3 codes.*

Description

Data frame contains the list of countries and their alpha-2 and aplha-3 codes.

Format

A data frame with 250 rows and 3 variables:

Country country names

Alpha2 countries' alpha-2 codes

Alpha3 countries' alpha-3 codes

oa2bibliometrix *Convert OpenAlex collection from data frame to bibliometrix object*

Description

It converts bibliographic collections gathered from OpenAlex database https://openalex.org/ into a bibliometrix data frame (https://bibliometrix.org/)

Usage

```
oa2bibliometrix(df)
```

Arguments

df	is bibliographic collection of works donwloaded from OpenALex.

Value

a data.frame with class "bibliometrix".

Examples

```
## Not run:

# Query to search all works citing the article:
#  Aria, M., & Cuccurullo, C. (2017). bibliometrix:
#   An R-tool for comprehensive science mapping analysis.
#   Journal of informetrics, 11(4), 959-975.

#  published in 2021.
#  The paper is associated to the OpenAlex id W2755950973.

#  Results have to be sorted by relevance score in a descending order.

query <- oa_query(
  identifier = NULL,
  entity = "works",
  cites = "W2755950973",
  from_publication_date = "2021-01-01",
  to_publication_date = "2021-12-31",
  search = NULL,
  endpoint = "https://api.openalex.org/"
)

res <- oa_request(
  query_url = query,
  count_only = FALSE,
  verbose = FALSE
)
```

```
df <- oa2df(res, entity = "works")

M <- oa2bibliometrix(df)

## End(Not run)
```

oa2df *Convert OpenAlex collection from list to data frame*

Description

It converts bibliographic collections gathered from OpenAlex database https://openalex.org/ into data frame. The function converts a collection of records about works, authors, institutions, venues or concepts obtained using `oa_request` into a data frame/tibble.

Usage

```
oa2df(
  data,
  entity,
  abstract = TRUE,
  count_only = FALSE,
  group_by = NULL,
  verbose = TRUE
)
```

Arguments

`data` is a list. data is the output of the function `oa_request`.

`entity` is a character. It indicates the scholarly entity of the search. The argument can be equal to entity = c("works", "authors", "venues", "institutions", "concepts"). The default value is entity = works".

`abstract` Logical. If TRUE, the function returns also the abstract of each item. Default is `abstract=TRUE`. The argument is ignored if entity is different from "works".

`count_only` Logical. If TRUE, the function returns only the number of item matching the query. Default is `count_only=FALSE`.

`group_by` Character. Property to group by. For example: "oa_status" for works. https://docs.openalex.org/api/get-groups-of-entities

`verbose` is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=TRUE`.

Value

a data.frame.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

# Query to search all works citing the article:
#  Aria, M., & Cuccurullo, C. (2017). bibliometrix:
#   An R-tool for comprehensive science mapping analysis.
#   Journal of informetrics, 11(4), 959-975.

#  published in 2021.
#  The paper is associated to the OpenAlex id W2755950973.

#  Results have to be sorted by relevance score in a descending order.

query <- oa_query(
  entity = "works",
  cites = "W2755950973",
  from_publication_date = "2021-01-01",
  to_publication_date = "2021-04-30"
)

res <- oa_request(
  query_url = query,
  count_only = FALSE,
  verbose = FALSE
)

oa2df(res, entity = "works")


## End(Not run)
```

`oaAuthors2df` *Convert OpenAlex collection of authors' records from list format to data frame*

Description

It converts bibliographic collection of authors' records gathered from OpenAlex database https://openalex.org/ into data frame. The function converts a list of authors' records obtained using `oa_request` into a data frame/tibble.

Usage

```
oaAuthors2df(data, verbose = TRUE)
```

Arguments

`data`	is a list. data is the output of the function `oa_request`.
`verbose`	is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=TRUE`.

Value

a data.frame.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

# Query to search information about all authors affiliated to the University of Naples Federico II
# which have authored at least 100 publications:

# University of Naples Federico II is associated to the OpenAlex id I71267560.


query_author <- oa_query(
  identifier = NULL,
  entity = "authors",
  last_known_institution.id = "I71267560",
  works_count = ">99"
)

res <- oa_request(
  query_url = query_author,
  count_only = FALSE,
  verbose = FALSE
)

df <- oa2df(res, entity = "authors")

df

## End(Not run)

# @export
```

`oaConcepts2df` *Convert OpenAlex collection of concepts' records from list format to data frame*

Description

It converts bibliographic collection of concepts' records gathered from OpenAlex database https://openalex.org/ into data frame. The function converts a list of concepts' records obtained using `oa_request` into a data frame/tibble.

Usage

```
oaConcepts2df(data, verbose = TRUE)
```

Arguments

`data`	is a list. data is the output of the function `oa_request`.
`verbose`	is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=TRUE`.

Value

a data.frame.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

# Query to search information about all Italian educational institutions


query_inst <- oa_query(
  entity = "concepts",
  display_name.search = "electrodynamics"
)

res <- oa_request(
  query_url = query_inst,
  count_only = FALSE,
  verbose = FALSE
)

df <- oa2df(res, entity = "concepts")

df

## End(Not run)

# @export
```

---

oaInstitutions2df *Convert OpenAlex collection of institutions' records from list format to data frame*

Description

It converts bibliographic collection of institutions' records gathered from OpenAlex database https://openalex.org/ into data frame. The function converts a list of institutions' records obtained using `oa_request` into a data frame/tibble.

Usage

```
oaInstitutions2df(data, verbose = TRUE)
```

Arguments

`data`	is a list. data is the output of the function `oa_request`.
`verbose`	is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=TRUE`.

Value

a data.frame.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

# Query to search information about all Italian educational institutions

query_inst <- oa_query(
  entity = "institutions",
  country_code = "it",
  type = "education"
)

res <- oa_request(
  query_url = query_inst,
  count_only = FALSE,
  verbose = FALSE
)

oa2df(res, entity = "institutions")

## End(Not run)

# @export
```

---

oaVenues2df *Convert OpenAlex collection of venues' records from list format to data frame*

Description

It converts bibliographic collection of venues' records gathered from OpenAlex database https://openalex.org/ into data frame. The function converts a list of venues' records obtained using `oa_request` into a data frame/tibble.

Usage

```
oaVenues2df(data, verbose = TRUE)
```

Arguments

`data`	is a list. data is the output of the function `oa_request`.
`verbose`	is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=TRUE`.

Value

a data.frame.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

# Query to search information about the Journal of Informetrics (OA id:V205292342)


query_inst <- oa_query(
  identifier = "V205292342",
  entity = "venues"
)

res <- oa_request(
  query_url = query_inst,
  count_only = FALSE,
  verbose = FALSE
)

df <- oa2df(res, entity = "venues")

df

## End(Not run)

# @export
```

`oaWorks2df` *Convert OpenAlex collection of works from list format to data frame*

Description

It converts bibliographic collection of works gathered from OpenAlex database https://openalex.org/ into data frame. The function converts a list of works obtained using `oa_request` into a data frame/tibble.

Usage

```
oaWorks2df(data, abstract = TRUE, verbose = TRUE)
```

Arguments

`data` is a list. data is the output of the function `oa_request`.

`abstract` Logical. If TRUE, the function returns also the abstract of each item. Default is `abstract=TRUE`.

`verbose` is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=TRUE`.

Value

a data.frame.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

# Query to search all works citing the article:
#  Aria, M., & Cuccurullo, C. (2017). bibliometrix:
#   An R-tool for comprehensive science mapping analysis.
#   Journal of informetrics, 11(4), 959-975.

#  published in 2021.
#  The paper is associated to the OpenAlex id W2755950973.

#  Results have to be sorted by relevance score in a descending order.

query <- oa_query(
  identifier = NULL,
  entity = "works",
  cites = "W2755950973",
  from_publication_date = "2021-01-01",
  to_publication_date = "2021-12-31",
  search = NULL,
  endpoint = "https://api.openalex.org/"
)

res <- oa_request(
  query_url = query,
  count_only = FALSE,
  verbose = FALSE
)

df <- oa2df(res, entity = "works")

df
```

```
## End(Not run)

# @export
```

oa_entities *Available entities in the OpenAlex database*

Description

Available entities in the OpenAlex database

Usage

```
oa_entities()
```

Value

Character vector of 5 entity options.

oa_fetch *A composition function to perform query building, requesting, and convert the result to a tibble/data frame.*

Description

A composition function to perform query building, requesting, and convert the result to a tibble/data frame.

Usage

```
oa_fetch(
  ...,
  identifier = NULL,
  entity = if (is.null(identifier)) NULL else id_type(identifier[[1]]),
  search = NULL,
  sort = NULL,
  group_by = NULL,
  output = c("tibble", "dataframe", "list"),
  abstract = TRUE,
  endpoint = "https://api.openalex.org/",
  per_page = 200,
  count_only = FALSE,
  mailto = NULL,
  verbose = FALSE
)
```

Arguments

`...` Filter arguments. Filters narrow the list down to just entities that meet a particular condition–specifically, a particular value for a particular attribute. Filters are formatted as attribute = value. The complete list of filter attributes for each entity can be found For example, ‘cited_by_count = ">100"‘, ‘title.search = c("bibliometric analysis", "science mapping")‘, or ‘to_publication_date = "2021-12-31"‘. at https://docs.openalex.org/api/get-lists-of-entities#filter

`identifier` Character. It indicates an item identifier.

`entity` Character. It indicates the scholarly entity of the search. The argument can be one of c("works", "authors", "venues", "institutions", "concepts"). If not provided, ‘entity‘ is guessed from ‘identifier‘.

`search` Character. Search is just another kind of filter, one that all five endpoints support. But unlike the other filters, search doesn’t require an exact match. To filter using search, append .search to the end of the property you’re filtering for.

`sort` Character. Property to sort by. For example: "display_name" for venues or "cited_by_count:desc" for works. See more at <https://docs.openalex.org/api/get-lists-of-entities/sort-entity-lists>.

`group_by` Character. Property to group by. For example: "oa_status" for works. https://docs.openalex.org/api/get-groups-of-entities

`output` Character. Type of output, either a list or a tibble/data.frame.

`abstract` Logical. If TRUE, the function returns also the abstract of each item. Default is `abstract = TRUE`. The argument is ignored if entity is different from "works".

`endpoint` is character. It indicates the url of the OpenAlex Endpoint API server. The default value is endpoint = "https://api.openalex.org/".

`per_page` is a numeric. It indicates how many items to download per page. The per-page argument can assume any number between 1 and 200. Default is `per_page=200`.

`count_only` is a logical. If TRUE, the function returns only the number of item matching the query. Default is `count_only=FALSE`.

`mailto` is a character. To get into the polite pool, the arguments mailto have to give OpenAlex an email where they can contact you.

`verbose` is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=FALSE`.

Value

A data.frame or a list. Result of the query.

Examples

```
## Not run:

paper_meta <- oa_fetch(
  identifier = "W2755950973",
  entity = "works",
  endpoint = "https://api.openalex.org/",
  count_only = TRUE,
  abstract = TRUE,
  verbose = TRUE
)

oa_fetch(
  entity = "works",
  doi = c(
    "10.1371/journal.pone.0266781",
    "10.1371/journal.pone.0267149"
  ),
  verbose = TRUE,
  count_only = TRUE
)

oa_fetch(
  entity = "works",
  doi = c(
    "10.1371/journal.pone.0266781",
    "10.1371/journal.pone.0267149"
  ),
  verbose = TRUE
)

## End(Not run)
```

oa_query *Generate an OpenAlex query from a set of parameters*

Description

It generates a valid query, written following the OpenAlex API Language, from a set of parameters.

Usage

```
oa_query(
  ...,
  identifier = NULL,
  entity = id_type(identifier[[1]]),
  search = NULL,
  sort = NULL,
  group_by = NULL,
  endpoint = "https://api.openalex.org/",
  verbose = FALSE
)
```

Arguments

`...` Filter arguments. Filters narrow the list down to just entities that meet a particular condition–specifically, a particular value for a particular attribute. Filters are formatted as attribute = value. The complete list of filter attributes for each entity can be found For example, ‘cited_by_count = ">100"‘, ‘title.search = c("bibliometric analysis", "science mapping")‘, or ‘to_publication_date = "2021-12-31"‘. at https://docs.openalex.org/api/get-lists-of-entities#filter

`identifier` Character. It indicates an item identifier.

`entity` Character. It indicates the scholarly entity of the search. The argument can be one of c("works", "authors", "venues", "institutions", "concepts"). If not provided, ‘entity‘ is guessed from ‘identifier‘.

`search` Character. Search is just another kind of filter, one that all five endpoints support. But unlike the other filters, search doesn’t require an exact match. To filter using search, append .search to the end of the property you’re filtering for.

`sort` Character. Property to sort by. For example: "display_name" for venues or "cited_by_count:desc" for works. See more at <https://docs.openalex.org/api/get-lists-of-entities/sort-entity-lists>.

`group_by` Character. Property to group by. For example: "oa_status" for works. https://docs.openalex.org/api/get-groups-of-entities

`endpoint` is character. It indicates the url of the OpenAlex Endpoint API server. The default value is endpoint = "https://api.openalex.org/".

`verbose` is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=FALSE`.

Value

a character containing the query in OpenAlex format.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

query_auth <- oa_query(identifier = "A923435168", verbose = TRUE)

### EXAMPLE 1: Full record about an entity.

# Query to obtain allinformation about a particular work/author/institution/etc.:

#  The following paper is associated to the OpenAlex-id W2755950973.

#  Aria, M., & Cuccurullo, C. (2017). bibliometrix:
#   An R-tool for comprehensive science mapping analysis.
#   Journal of informetrics, 11(4), 959-975.

query_work <- oa_query(
  identifier = "W2755950973",
  endpoint = "https://api.openalex.org/",
  verbose = TRUE
)


#  The author Massimo Aria is associated to the OpenAlex-id A923435168:

query_auth <- oa_query(identifier = "A923435168", verbose = TRUE)


### EXAMPLE 2: all works citing a particular work.

# Query to search all works citing the article:
#  Aria, M., & Cuccurullo, C. (2017). bibliometrix:
#   An R-tool for comprehensive science mapping analysis.
#   Journal of informetrics, 11(4), 959-975.

#  published in 2021.
#  The paper is associated to the OpenAlex id W2755950973.

#  Results have to be sorted by relevance score in a descending order.

query1 <- oa_query(
  entity = "works",
  cites = "W2755950973",
  from_publication_date = "2021-01-01",
  to_publication_date = "2021-12-31",
  verbose = TRUE
)

### EXAMPLE 3: All works matching a string in their title

# Query to search all works containing the exact string
# "bibliometric analysis" OR "science mapping" in the title, published in the first half of 2021.

# Results have to be sorted by relevance score in a descending order.

query2 <- oa_query(
  entity = "works",
  title.search = c("bibliometric analysis", "science mapping"),
  from_publication_date = "2021-01-01",
  to_publication_date = "2021-06-30",
  sort = "cited_by_count:desc",
  verbose = TRUE
)

## End(Not run)
```

`oa_random` *oa_fetch but for a random query*

Description

oa_fetch but for a random query

Usage

```
oa_random(
  entity = oa_entities(),
  output = c("tibble", "dataframe", "list"),
  endpoint = "https://api.openalex.org/"
)
```

Arguments

`entity`	Character. It indicates the scholarly entity of the search. The argument can be one of c("works", "authors", "venues", "institutions", "concepts"). If not provided, ‘entity‘ is guessed from ‘identifier‘.
`output`	Character. Type of output, either a list or a tibble/data.frame.
`endpoint`	is character. It indicates the url of the OpenAlex Endpoint API server. The default value is endpoint = "https://api.openalex.org/".

Value

A data.frame or a list. One row or one element. Result of the random query.

Examples

```
oa_random()
```

`oa_request` *Get bibliographic records from OpenAlex databases*

Description

It gets bibliographic records from OpenAlex database https://openalex.org/. The function `oa_request` queries OpenAlex database using a query formulated through the function `oa_query`.

Usage

```
oa_request(
  query_url,
  per_page = 200,
  count_only = FALSE,
  mailto = oa_email(),
  verbose = FALSE
)
```

Arguments

`query_url` is a character. It contains a search query formulated using the OpenAlex API language. A query can be automatically generated using the function `oa_query`.

`per_page` is a numeric. It indicates how many items to download per page. The per-page argument can assume any number between 1 and 200. Default is `per_page=200`.

`count_only` is a logical. If TRUE, the function returns only the number of item matching the query. Default is `count_only=FALSE`.

`mailto` is a character. To get into the polite pool, the arguments mailto have to give OpenAlex an email where they can contact you.

`verbose` is a logical. If TRUE, information about the querying process will be plotted on screen. Default is `verbose=FALSE`.

Value

a data.frame or a list.

For more extensive information about OpenAlex API, please visit: https://docs.openalex.org/api

Examples

```
## Not run:

### EXAMPLE 1: Full record about an entity.

# Query to obtain all information about a particular work/author/institution/etc.:

#  The following paper is associated to the OpenAlex-id W2755950973.

#  Aria, M., & Cuccurullo, C. (2017). bibliometrix:
#   An R-tool for comprehensive science mapping analysis.
#   Journal of informetrics, 11(4), 959-975.


query_work <- oa_query(
  identifier = "W2755950973",
  entity = "works",
  endpoint = "https://api.openalex.org/"
)

res <- oa_request(
  query_url = query_work,
  count_only = FALSE,
  verbose = FALSE
)

#  The author Massimo Aria is associated to the OpenAlex-id A923435168.


query_author <- oa_query(
  identifier = "A923435168",
  entity = "authors",
  endpoint = "https://api.openalex.org/"
)

res <- oa_request(
  query_url = query_author,
  count_only = FALSE,
  verbose = FALSE
)



### EXAMPLE 2: all works citing a particular work.

# Query to search all works citing the article:
#  Aria, M., & Cuccurullo, C. (2017). bibliometrix:
#   An R-tool for comprehensive science mapping analysis.
#   Journal of informetrics, 11(4), 959-975.

#  published in 2021.
#  The paper is associated to the OpenAlex id W2755950973.

#  Results have to be sorted by relevance score in a descending order.

query2 <- oa_query(
  identifier = NULL,
  entity = "works",
  filter = "cites:W2755950973",
  from_publication_date = "2021-01-01",
  to_publication_date = "2021-12-31",
  search = NULL,
  endpoint = "https://api.openalex.org/"
)

res2 <- oa_request(
  query_url = query2,
  count_only = FALSE,
  verbose = FALSE
)

### EXAMPLE 3: All works matching a string in their title

# Query to search all works containing the exact string
# "bibliometric analysis" OR "science mapping" in the title, published in 2020 or 2021.

# Results have to be sorted by relevance score in a descending order.


query3 <- oa_query(
  identifier = NULL,
  entity = "works",
  filter = 'title.search:"bibliometric analysis"|"science mapping"',
  from_publication_date = "2020-01-01",
  to_publication_date = "2021-12-31",
  search = NULL,
  endpoint = "https://api.openalex.org/"
)

res3 <- oa_request(
  query_url = query3,
  count_only = FALSE,
  verbose = FALSE
)

### EXAMPLE 4: How to check how many works match a query
# Query to search all works containing the exact string
# "bibliometric analysis" OR "science mapping" in the title, published in 2020 or 2021.

# Query only to know how many works could be retrieved (count_only=TRUE)

query4 <- oa_query(
  identifier = NULL,
  entity = "works",
  filter = 'title.search:"bibliometric analysis"|"science mapping"',
  from_publication_date = "2020-01-01",
  to_publication_date = "2021-12-31",
  search = NULL,
  endpoint = "https://api.openalex.org/"
)

res4 <- oa_request(
  query_url = query4,
  count_only = TRUE,
  verbose = FALSE
)

res4$count # number of items retrieved by our query

## End(Not run)
```

oa_snowball *A function to perform a snowball search and convert the result to a tibble/data frame.*

Description

A function to perform a snowball search and convert the result to a tibble/data frame.

Usage

```
oa_snowball(
  identifier = NULL,
  output = c("tibble", "dataframe"),
  mailto = NULL,
  endpoint = "https://api.openalex.org/",
  verbose = FALSE
)
```

Arguments

identifier	Character. It indicates a vector of item identifiers.
output	a tibble/data.frame.
mailto	is a character. To get into the polite pool, the arguments mailto have to give OpenAlex an email where they can contact you.
endpoint	is character. It indicates the url of the OpenAlex Endpoint API server. The default value is endpoint = "https://api.openalex.org/".
verbose	is a logical. If TRUE, information about the querying process will be plotted on screen. Default is verbose=FALSE.

Value

A data.frame or a tibble. Result of the snowball search.

Examples

```
## Not run:

snowball_docs <- oa_snowball(
  identifier = c("W2741809807", "W2755950973"),
  endpoint = "https://api.openalex.org/",
  verbose = TRUE
)

## End(Not run)
```

show_authors *Simplify the OpenAlex authors result*

Description

This function is mostly for the package's internal use, but we export it so you can try it out. However, we expect that you'll likely write your own function to simplify the result however you want.

Usage

```
show_authors(x, simp_func = utils::head)
```

Arguments

`x`	Dataframe/tibble. Result of the OpenAlex query for authors already converted to dataframe/tibble.
`simp_func`	R function to simplify the result. Default to ‘head‘. If you want the entire table, set ‘simp_fun = identity‘

Value

Simplified tibble to display.

Examples

```
show_authors(oa_fetch(
  identifier = c("A923435168", "A2208157607"),
  verbose = TRUE
))
```

`show_works` *Simplify the OpenAlex works result*

Description

This function is mostly for the package’s internal use, but we export it so you can try it out. However, we expect that you’ll likely write your own function to simplify the result however you want.

Usage

```
show_works(x, simp_func = utils::head)
```

Arguments

`x`	Dataframe/tibble. Result of the OpenAlex query for authors already converted to dataframe/tibble.
`simp_func`	R function to simplify the result. Default to ‘head‘. If you want the entire table, set ‘simp_fun = identity‘.

Value

Simplified tibble to display.

Examples

```
show_works(oa_fetch(
  identifier = c("W2741809807", "W2755950973"),
  verbose = TRUE
))
```

Index