Package: wosr (via r-universe)

August 28, 2024

Type Package

Title Clients to the 'Web of Science' and 'InCites' APIs

Description R clients to the 'Web of Science' and 'InCites' <https://clarivate.com/products/data-integration/> APIs, which allow you to programmatically download publication and citation data indexed in the 'Web of Science' and 'InCites' databases.

URL https://vt-arc.github.io/wosr/index.html

BugReports https://github.com/vt-arc/wosr/issues

Version 0.3.0

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.1)

Imports httr, xml2, jsonlite, pbapply, utils, tools

RoxygenNote 6.1.0

Suggests testthat, knitr, rmarkdown, dplyr

NeedsCompilation no

Author Christopher Baker [aut, cre]

Maintainer Christopher Baker <chriscrewbaker@gmail.com>

Date/Publication 2018-11-02 05:30:03 UTC

Repository https://crew102.r-universe.dev

RemoteUrl https://github.com/cran/wosr

RemoteRef HEAD

RemoteSha 0642c3382e6e3e2d487bb0eed9f787c45f7f47a6

Contents

`auth` *Authenticate user credentials*

Description

`auth` asks the API's server for a session ID (SID), which you can then pass along to either `query_wos` or `pull_wos`. Note, there are limits on how many session IDs you can get in a given period of time (roughly 5 SIDs in a 5 minute period).

Usage

```
auth(username = Sys.getenv("WOS_USERNAME"),
  password = Sys.getenv("WOS_PASSWORD"))
```

Arguments

`username`	Your username. Specify `username = NULL` if you want to use IP-based authentication.
`password`	Your password. Specify `password = NULL` if you want to use IP-based authentication.

Value

A session ID

Examples

```
## Not run:

# Pass user credentials in manually:
auth(”some_username”, password = ”some_password”)

# Use the default of looking for username and password in envvars, so you
# don't have to keep specifying them in your code:
Sys.setenv(WOS_USERNAME = ”some_username”, WOS_PASSWORD = ”some_password”)
auth()

## End(Not run)
```

create_ut_queries *Create a vector of UT-based queries*

Description

Use this function when you have a bunch of UTs whose data you want to pull and you need to write a series of UT-based queries to do so (i.e., queries in the form "UT = (WOS:000186387100005 OR WOS:000179260700001)").

Usage

```
create_ut_queries(uts, uts_per_query = 200)
```

Arguments

`uts`	UTs that will be placed inside the UT-based queries.
`uts_per_query`	Number of UTs to include in each query. Note, there is a limit on how long your query can be, so you probably want to keep this set to around 200.

Value

A vector of queries. You can feed these queries to `pull_wos_apply` to download data for each query.

Examples

```
## Not run:

data <- pull_wos('TS = (”animal welfare”) AND PY = (2002-2003)')
queries <- create_ut_queries(data$publication$ut)
pull_wos_apply(queries)

## End(Not run)
```

pull_cited_refs *Pull cited references*

Description

Pull cited references

Usage

```
pull_cited_refs(uts, sid = auth(Sys.getenv("WOS_USERNAME"),
  Sys.getenv("WOS_PASSWORD")), ...)
```

Arguments

`uts` Vector of UTs (i.e., publications) whose cited references you want.

`sid` Session identifier (SID). The default setting is to get a fresh SID each time you query WoS via a call to `auth`. However, you should try to reuse SIDs across queries so that you don't run into the throttling limits placed on new sessions.

`...` Arguments passed along to `POST`.

Value

A data frame with the following columns:

ut The publication that is doing the citing. These are the UTs that you submitted to `pull_cited_refs`. If one of your publications doesn't have any cited refs, it will not appear in this column.

doc_id The cited ref's document identifier (similar to a UT).

title Roughly equivalent to the cited ref's title.

journal Roughly equivalent to the cited ref's journal.

author The cited ref's first author.

tot_cites The total number of citations the cited ref has received.

year The cited ref's publication year.

page The cited ref's page number.

volume The cited ref's journal volume.

Examples

```
## Not run:

sid <- auth("your_username", password = "your_password")
uts <- c("WOS:000362312600021", "WOS:000439855300030", "WOS:000294946900020")
pull_cited_refs(uts, sid)

## End(Not run)
```

`pull_incites` *Pull data from the InCites API*

Description

Important note: The throttling limits on the InCites API are not documented anywhere and are difficult to determine from experience. As such, whenever `pull_incites` receives a throttling error from the server, it uses exponential backoff (with a maximum wait time of 45 minutes) to determine how long to wait before retrying.

Usage

```
pull_incites(uts, key = Sys.getenv("INCITES_KEY"), as_raw = FALSE, ...)
```

Arguments

`uts`	A vector of UTs whose InCites data you would like to download. Each UT is a 15-digit identifier for a given publication. You can specify the UT using only these 15 digits or you can append the 15 digits with "WOS:" (e.g., "000346263300011" or "WOS:000346263300011").
`key`	The developer key that the server will use for authentication.
`as_raw`	Do you want the data frame that is returned by the API to be returned to you in its raw form? This option can be useful if the API has changed the format of the data that it is serving, in which case specifying `as_raw` = `TRUE` may avoid an error that would otherwise occur during `pull_incites`'s data processing step.
`...`	Arguments passed along to `GET`.

Value

A data frame where each row corresponds to a different publication. The definitions for the columns in this data frame can be found online at the API's documentation page (see the `DocumentLevelMetricsByUT` method details for definitions). Note that the column names are all converted to lowercase by `pull_incites` and the 0/1 flag variables are converted to booleans. Also note that not all publications indexed in WoS are also indexed in InCites, so you may not get data back for some UTs.

Examples

```
## Not run:

uts <- c(
  "WOS:000346263300011", "WOS:000362312600021", "WOS:000279885800004",
  "WOS:000294667500003", "WOS:000294946900020", "WOS:000412659200006"
)
pull_incites(uts, key = "some_key")

pull_incites(c("000346263300011", "000362312600021"), key = "some_key")

## End(Not run)
```

pull_related_recs *Pull related records*

Description

Pull the records that have at least one citation in common with a publication of interest.

Usage

```
pull_related_recs(uts, num_recs, editions = c("SCI", "SSCI", "AHCI",
  "ISTP", "ISSHP", "BSCI", "BHCI", "IC", "CCR", "ESCI"),
  sid = auth(Sys.getenv("WOS_USERNAME"), Sys.getenv("WOS_PASSWORD")),
  ...)
```

Arguments

`uts`	The documents whose related records you want to pull.
`num_recs`	Number of related records to pull for each UT. This value must be <= 100.
`editions`	Web of Science editions to query. Possible values are listed here.
`sid`	Session identifier (SID). The default setting is to get a fresh SID each time you query WoS via a call to `auth`. However, you should try to reuse SIDs across queries so that you don't run into the throttling limits placed on new sessions.
`...`	Arguments passed along to `POST`.

Value

A data frame with the following columns:

ut The publications that you passed into `pull_related_recs`. If one of your publications doesn't have any related records, it won't appear here.

related_rec The publication that is related to `ut`.

rec_num The related record's ordering in the result set returned by the API. Records that share more citations with your UTs will have smaller `rec_nums`.

Examples

```
## Not run:

sid <- auth("your_username", password = "your_password")
uts <- c("WOS:000272877700013", "WOS:000272366800025")
out <- pull_related_recs(uts, 5, sid = sid)

## End(Not run)
```

pull_wos *Pull data from the Web of Science*

Description

pull_wos wraps the process of querying, downloading, parsing, and processing Web of Science data.

Usage

```
pull_wos(query, editions = c("SCI", "SSCI", "AHCI", "ISTP", "ISSHP",
  "BSCI", "BHCI", "IC", "CCR", "ESCI"),
  sid = auth(Sys.getenv("WOS_USERNAME"), Sys.getenv("WOS_PASSWORD")),
  ...)
```

Arguments

`query` Query string. See the WoS query documentation page for details on how to write a query as well as this list of example queries.

`editions` Web of Science editions to query. Possible values are listed here.

`sid` Session identifier (SID). The default setting is to get a fresh SID each time you query WoS via a call to `auth`. However, you should try to reuse SIDs across queries so that you don't run into the throttling limits placed on new sessions.

`...` Arguments passed along to `POST`.

Value

A list of the following data frames:

publication A data frame where each row corresponds to a different publication. Note that each publication has a distinct `ut`. There is a one-to-one relationship between a `ut` and each of the columns in this table.

author A data frame where each row corresponds to a different publication/author pair (i.e., a `ut`/`author_no` pair). In other words, each row corresponds to a different author on a publication. You can link the authors in this table to the `address` and `author_address` tables to get their addresses (if they exist). See example in FAQs for details.

address A data frame where each row corresponds to a different publication/address pair (i.e., a `ut`/`addr_no` pair). In other words, each row corresponds to a different address on a publication. You can link the addresses in this table to the `author` and `author_address` tables to see which authors correspond to which addresses. See example in FAQs for details.

author_address A data frame that specifies which authors correspond to which addresses on a given publication. This data frame is meant to be used to link the `author` and `address` tables together.

jsc A data frame where each row corresponds to a different publication/jsc (journal subject category) pair. There is a many-to-many relationship between `ut`'s and `jsc`'s.

keyword A data frame where each row corresponds to a different publication/keyword pair. These are the author-assigned keywords.

keywords_plus A data frame where each row corresponds to a different publication/keywords_plus pair. These keywords are the keywords assigned by Clarivate Analytics through an automated process.

grant A data frame where each row corresponds to a different publication/grant agency/grant ID triplet. Not all publications acknowledge a specific grant number in the funding acknowledgement section, hence the `grant_id` field can be `NA`.

doc_type A data frame where each row corresponds to a different publication/document type pair.

Examples

```
## Not run:

sid <- auth("your_username", password = "your_password")
pull_wos("TS = (dog welfare) AND PY = 2010", sid = sid)

# Re-use session ID. This is best practice to avoid throttling limits:
pull_wos("TI = \"dog welfare\"", sid = sid)

# Get fresh session ID:
pull_wos("TI = \"pet welfare\"", sid = auth("your_username", "your_password"))

# It's best to see how many records your query matches before actually
# downloading the data. To do this, call query_wos before running pull_wos:
query <- "TS = ((cadmium AND gill*) NOT Pisces)"
query_wos(query, sid = sid) # shows that there are 1,611 matching publications
pull_wos(query, sid = sid)

## End(Not run)
```

`pull_wos_apply` *Run* `pull_wos` *across multiple queries*

Description

Run `pull_wos` across multiple queries

Usage

```
pull_wos_apply(queries, editions = c("SCI", "SSCI", "AHCI", "ISTP",
  "ISSHP", "BSCI", "BHCI", "IC", "CCR", "ESCI"),
  sid = auth(Sys.getenv("WOS_USERNAME"), Sys.getenv("WOS_PASSWORD")),
  ...)
```

Arguments

queries — Vector of queries to issue to the WoS API and pull data for.

editions — Web of Science editions to query. Possible values are listed here.

sid — Session identifier (SID). The default setting is to get a fresh SID each time you query WoS via a call to auth. However, you should try to reuse SIDs across queries so that you don't run into the throttling limits placed on new sessions.

... — Arguments passed along to POST.

Value

The same set of data frames that pull_wos returns, with the addition of a data frame named query. This data frame frame tells you which publications were returned by a given query.

Examples

```
## Not run:

queries <- c('TS = "dog welfare"', 'TS = "cat welfare"')
# we can name the queries so that these names appear in the queries data
# frame returned by pull_wos_apply():
names(queries) <- c("dog welfare", "cat welfare")
pull_wos_apply(queries)

## End(Not run)
```

---

query_wos *Query the Web of Science*

Description

Returns the number of records that match a given query. It's best to call this function before calling pull_wos so that you know how many records you're trying to download before attempting to do so.

Usage

```
query_wos(query, editions = c("SCI", "SSCI", "AHCI", "ISTP", "ISSHP",
  "BSCI", "BHCI", "IC", "CCR", "ESCI"),
  sid = auth(Sys.getenv("WOS_USERNAME"), Sys.getenv("WOS_PASSWORD")),
  ...)
```

Arguments

query — Query string. See the WoS query documentation page for details on how to write a query as well as this list of example queries.

editions — Web of Science editions to query. Possible values are listed here.

`sid` Session identifier (SID). The default setting is to get a fresh SID each time you query WoS via a call to `auth`. However, you should try to reuse SIDs across queries so that you don't run into the throttling limits placed on new sessions.

`...` Arguments passed along to `POST`.

Value

An object of class `query_result`. This object contains the number of publications that are returned by your query (`rec_cnt`), as well as some info that `pull_wos` uses when it calls `query_wos` internally.

Examples

```
## Not run:

# Get session ID and reuse it across queries:
sid <- auth("some_username", password = "some_password")

query_wos("TS = (\"dog welfare\") AND PY = (1990-2007)", sid = sid)

# Finds records in which Max Planck appears in the address field.
query_wos("AD = Max Planck", sid = sid)

# Finds records in which Max Planck appears in the same address as Mainz
query_wos("AD = (Max Planck SAME Mainz)", sid = sid)

## End(Not run)
```

`query_wos_apply` *Run `query_wos` across multiple queries*

Description

Run `query_wos` across multiple queries

Usage

```
query_wos_apply(queries, editions = c("SCI", "SSCI", "AHCI", "ISTP",
  "ISSHP", "BSCI", "BHCI", "IC", "CCR", "ESCI"),
  sid = auth(Sys.getenv("WOS_USERNAME"), Sys.getenv("WOS_PASSWORD")),
  ...)
```

Arguments

`queries` Vector of queries run.

`editions` Web of Science editions to query. Possible values are listed here.

`sid` Session identifier (SID). The default setting is to get a fresh SID each time you query WoS via a call to `auth`. However, you should try to reuse SIDs across queries so that you don't run into the throttling limits placed on new sessions.

`...` Arguments passed along to `POST`.

Value

A data frame which lists the number of records returned by each of your queries.

Examples

```
## Not run:

queries <- c('TS = "dog welfare"', 'TS = "cat welfare"')
query_wos_apply(queries)

## End(Not run)
```

---

read_wos_data *Read WoS data*

Description

Reads in a series of CSV files (which were written via write_wos_data) and places the data in an object of class wos_data.

Usage

```
read_wos_data(dir)
```

Arguments

dir	Path to the directory where you wrote the CSV files.

Value

An object of class wos_data.

Examples

```
## Not run:

sid <- auth("your_username", password = "your_password")
wos_data <- pull_wos("TS = (dog welfare) AND PY = 2010", sid = sid)

# Write files to working directory
write_wos_data(wos_data, ".")
# Read data back into R
wos_data <- read_wos_data(".")

## End(Not run)
```

wosr *wosr*

Description

wosr

write_wos_data *Write WoS data*

Description

Writes each of the data frames in an object of class `wos_data` to its own csv file.

Usage

```
write_wos_data(wos_data, dir)
```

Arguments

`wos_data`	An object of class `wos_data`, created by calling `pull_wos`.
`dir`	Path to the directory where you want to write the files. If the directory doesn't yet exist, `write_wos_data` will create it for you. Note, this directory cannot already have WoS data files in it.

Value

Nothing. Files are written to disk.

Examples

```
## Not run:

sid <- auth("your_username", password = "your_password")
wos_data <- pull_wos("TS = (dog welfare) AND PY = 2010", sid = sid)

# Write files to working directory
write_wos_data(wos_data, ".")

# Write files to "wos-data" dir
write_wos_data(wos_data, "wos-data")

## End(Not run)
```

Index