

Package ‘wosr’

September 27, 2018

Type Package

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

Description An R client to the 'Web of Science' and 'InCites' <<https://clarivate.com/products/data-integration/>> web services, 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 0.2.0

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>

Repository CRAN

Date/Publication 2018-09-27 13:40:02 UTC

R topics documented:

auth	2
pull_incites	3
pull_wos	4
pull_wos_apply	5
query_wos	6

query_wos_apply	7
read_wos_data	8
wosr	9
write_wos_data	9

Index	11
--------------	-----------

auth	<i>Authenticate user credentials</i>
------	--------------------------------------

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 time 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 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)
```

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_wos

*Pull data from the WoS API***Description**

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

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 a the 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 the WoS, via a call to auth . However, you should try to reuse SID values over multiple queries so that you don't run into the throttling limits placed on new sessions.
...	Arguments passed along to POST .

Value

A list of data frames, which can be joined together like a relational database:

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 fields 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 vignette 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 vignette 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 keywords 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 the Web of Science 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 acknowledgment 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 the WoS, via a call to auth . However, you should try to reuse SID values over multiple 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 contains a mapping of queries to publications that were returned by those queries.

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 are trying to download before you attempt 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 a the 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 the WoS, via a call to auth . However, you should try to reuse SID values over multiple 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 has 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	<i>Run query_wos across multiple queries</i>
-----------------	--

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 to issue to the WoS API.
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 the WoS, via a call to auth . However, you should try to reuse SID values over multiple 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 the queries in your queries vector.

Examples

```
## Not run:  
  
queries <- c('TS = "dog welfare"', 'TS = "cat welfare"')  
query_wos_apply(queries)  
  
## End(Not run)
```

read_wos_data	<i>Read WoS data</i>
---------------	----------------------

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 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. Each file will be named after the name of the data frame.

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.

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
dir.create("wos-data")
write_wos_data(wos_data, "wos-data")

## End(Not run)
```

Index

auth, [2](#), [4](#), [6–8](#)

GET, [3](#)

POST, [4](#), [6–8](#)

pull_incites, [3](#)

pull_wos, [2](#), [4](#), [6](#), [7](#), [9](#)

pull_wos_apply, [5](#)

query_wos, [2](#), [6](#)

query_wos_apply, [7](#)

read_wos_data, [8](#)

wosr, [9](#)

wosr-package (wosr), [9](#)

write_wos_data, [8](#), [9](#)