Package ‘PAWR’

January 17, 2020

Title Pushshift API Wrapper for R

Version 0.0.1

Description Easily scrape reddit content using the pushshift.io API. PAWR formats your queries, takes breaks when exceeding the rate limit, and outputs the data in a usable format.

BugReports https://github.com/Spiritspeak/PAWR/issues

Depends R (¿= 3.6.1), magrittr, dplyr, httr

Imports rvest

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

ByteCompile true

R topics documented:

GetPSData *Query data from pushshift.io*

Description

Query data from pushshift.io

Usage

```
GetPSData(type = c("comment", "submission", "subreddit"), as.df = T,
  purge = F, verbose = getOption("PAWR.VerboseGet"), size = 500,
  aggs = c("none", "author", "link_id", "created_utc", "subreddit"),
  agg_size = 0, q = NULL, metadata = TRUE, ...)
```

Arguments

`type` Type of requested content. Can be `comment`, `submission`, or `subreddit`.

`as.df` Convert output to `data.frame`? Defaults to `TRUE`.

`purge` Purge deleted posts? Defaults to `FALSE`.

`verbose` Should output be verbose? Defaults to a global option which can be set with `options(PAWR.VerboseGet=TRUE/FALSE)`.

`size` Maximum number of pieces of content to return. Defaults to the maximum, which is 500, except when using aggs, when it defaults to 0.

`agg_size` Maximum number of values to return when using aggs; defaults to 500 unless you're not using aggs.

`q` Query term.

`metadata` Request metadata from pushshift, which will be used to check whether the query was successful or some of pushshift's shards failed to respond. This is recommended for academic research.

`...` Other valid parameters. Run `PSParams()` to see all valid parameters and their descriptions.

Value

If `as,df=T`, returns a data.frame; else, returns a list.

Examples

```
#Get u/spez's first ever comment
GetPSData(author="spez",after=0,size=1)

#See in which subreddits the word "gamer" is used the most
GetPSData(q="Gamer",aggs="subreddit")
```

PaginateAggs *Paginate aggs*

Description

Send multiple queries to pushshift.io to get all available information for your request. This function is meant to get around the maximum of 1000 items returned by a single aggs query.

Usage

```
PaginateAggs(type = c("comment", "submission", "subreddit"),
  aggs = c("author", "link_id", "created_utc", "subreddit"),
  paginate_by = c("date", "author"),
  before = round(as.numeric(Sys.time())), after = NULL,
  timescope = NULL, verbose = getOption("PAWR.VerbosePaginate"), ...)
```

Arguments

type — Type of requested content. Can be `comment`, `submission`, or `subreddit`.

aggs — What should be aggregated over?

paginate_by — Define which variable should be used to break the data into smaller chunks; either `author` or `date`.

verbose — Should output be verbose? Defaults to a global option which can be set with `options(PAWR.VerbosePaginate=TRUE/FALSE)`.

... — Other valid parameters. Run `PSParams()` to see all valid parameters and their descriptions.

Examples

```
#Find out on which subreddits the users of r/cheese post
#Analysis is limited to December 2019
users<-PaginateAggs(aggs="author",paginate_by="date",
  subreddit="cheese",timescope=30*24*60*60,before=1577836800)
users<-users$key
#remove bots and missing values
users<-users[!(users %in% c("[deleted]","AutoModerator"))]
#Posting behavior of all authors is aggregated.
subreddits<-PaginateAggs(aggs="subreddit",paginate_by="author",
  author=users,timescope=30*24*60*60,before=1577836800)
```

PaginateData *Paginate PushShift Data*

Description

Send multiple queries to pushshift.io, to be able to get all data within a given date range.

Usage

```
PaginateData(type = c("comment", "submission", "subreddit"),
  verbose = getOption("PAWR.VerbosePaginate"),
  before = round(as.numeric(Sys.time())), after = NULL,
  timescope = NULL, ...)
```

Arguments

type — Type of requested content. Can be `comment`, `submission`, or `subreddit`.

verbose — Should output be verbose? Defaults to a global option which can be set with `options(PAWR.VerbosePaginate=TRUE/FALSE)`.

before — Upper limit in the date range of posts to be fetched.

after — Lower limit in the date range of posts to be fetched.

timescope — Time range, in seconds, within which posts should be fetched; works in conjunction with either `before` or `after`. When argument `after` is used, `timescope` causes the current function to fetch data that was created up to N seconds after the timestamp defined in `after` When argument `before` is used, `timescope` causes the current function to fetch data that was created up to N seconds before the timestamp defined in `before`

... — Other valid parameters. Run `PSParams()` to see all valid parameters and their descriptions.

Examples

```
#Get all comments from today containing the word "chocolate"
PaginateData(timescope=24 * 60 * 60,q="chocolate")
```

PAWR *PAWR: The Pushshift API Wrapper for R.*

Description

PAWR: The Pushshift API Wrapper for R.

Package options

- `PAWR.VerboseGet` designates whether the main data retrieval function should produce verbose output. Useful when debugging.
- `PAWR.VerbosePaginate` designates whether pagination functions should produce verbose output.
- `PAWR.UserAgent` is the useragent used by PAWR when querying pushshift.io.

UtilityFunctions *Utility Functions*

Description

Utility Functions

Usage

```
refreshPAWR(verbose = T)

list2df(li)

unevenrbind(df1, df2)

PSParams(type = c("all", "comment", "submission", "subreddit"))
```

Arguments

`li`	List of lists, to be converted to `data.frame`.
`df1, df2`	To-be-merged data frames with an uneven number of columns and/or nonmatching column names
`type`	Character. The type of content that parameters are being looked up for (comment, submission, subreddit). Defaults to all.

Details

`refreshPAWR()` re-fetches the rate limit and parameter list from pushshift.io

Index