Title  An API Client for Wikimedia Traffic Data
Version  0.3.0
Date  2016-10-16
Author  Oliver Keyes [aut, cre], Jeremiah Lewis [ctb]
Maintainer  Oliver Keyes <ironholds@gmail.com>
Description  Pageview data from the 'Wikimedia' sites, such as
 'Wikipedia' <https://www.wikipedia.org/>, from entire projects to per-article
 levels of granularity, through the new RESTful API and data source <https://
 wikimedia.org/api/rest_v1/?doc>.
License  MIT + file LICENSE
LazyData  true
URL  https://github.com/ironholds/pageviews
BugReports  https://github.com/ironholds/pageviews
Suggests  testthat, knitr, rmarkdown, WikipediR, WikidataR
Imports  jsonlite, httr, curl
VignetteBuilder  knitr
RoxygenNote  5.0.1
NeedsCompilation  no
Repository  CRAN
Date/Publication  2016-10-17 11:09:34

R topics documented:

article_pageviews ...................................................... 2
pageviews ............................................................. 3
pageview_timestamps ................................................... 3
project_pageviews ..................................................... 4
top_articles ............................................................ 5
Index ........................................................................ 6
article_pageviews  

Retrieves Pageview Data for an Article

Description
retrieves the pageview data for a particular article on a project, within a provided time-range.

Usage
article_pageviews(project = "en.wikipedia",  
article = "R (programming language)", platform = "all",  
user_type = "all", start = "2015010100", end = NULL, reformat = TRUE,  
...)

Arguments
- **project**: the name of the project, structured as [language_code].[project] (see the default).
- **article**: the article(s) you want to retrieve data for. Ideally features underscores in the title instead of spaces, but happily converts if you forget to do this.
- **platform**: The platform the pageviews came from; One or more of "all", "desktop", "mobile-web" and "mobile-app". Set to "all" by default.
- **user_type**: the type of users. One or more of "all", "user", "spider" or "bot". "all" by default.
- **start**: the start YYYYMMDDHH of the range you want to cover. This can easily be grabbed from R date/time objects using pageview_timestamps.
- **end**: the end YYYYMMDDHH of the range you want to cover. NULL by default, meaning that it returns 1 day of data.
- **reformat**: Whether to reformat the results as a data.frame or not. TRUE by default.
- **...**: further arguments to pass to httr’s GET.

See Also
top_articles for the top articles per project in a given date range, and project_pageviews for per-project pageviews.

Examples
# Basic example  
r_pageviews <- article_pageviews()

# Modify the article  
obama_pageviews <- article_pageviews(article = "Barack_Obama")
Description

Pageview data from the 'Wikimedia' sites, such as Wikipedia (https://www.wikipedia.org/), from entire projects to by-article levels of granularity.

Description

pageview_timestamps converts Date and POSIXlt and ct objects to work nicely with the start and end parameters in pageviews functions.

Usage

pageview_timestamps(timestamps = Sys.Date(), first = TRUE)

Arguments

timestamps a vector of character, Date, POSIXlt or POSIXct objects.
first whether to, if timestamps is of date objects, assume the first hour in a day (TRUE) or the last (FALSE). TRUE by default.

Value

a character vector containing timestamps that can be used with article_pageviews et al.

See Also

article_pageviews and project_pageviews, where you can make use of this function.

Examples

# Using a Date
pageview_timestamps(Sys.Date())

# Using a POSIXct object
pageview_timestamps(Sys.time())

# Validate a character string
pageview_timestamps("2016020800")
project_pageviews  Retrieve Per-Project Pageview Counts

Description

Retrieve pageview counts for a particular project.

Usage

project_pageviews(project = "en.wikipedia", platform = "all",
user_type = "all", granularity = "daily", start = "2015100100",
end = NULL, reformat = TRUE, ...)

Arguments

project  the name of the project, structured as [language_code].[project] (see the default).
platform  The platform the pageviews came from; one or more of "all", "desktop", "mobile-web" and "mobile-app". Set to "all" by default.
user_type  the type of users. one or more of "all", "user", "spider" or "bot". "all" by default.
granularity  the granularity of data to return; do you want hourly or daily counts? Set to "daily" by default.
start  the start YYYYMMDDHH of the range you want to cover. This can be easily grabbed from R date/time objects using pageview_timestamps
end  the end YYYYMMDDHH of the range you want to cover. NULL by default, meaning that it returns 1 day/hour of data (depending on the value passed to granularity).
reformat  Whether to reformat the results as a data.frame or not. TRUE by default.
...  further arguments to pass to httr's GET.

See Also

top_articles for the top articles per project in a given date range, and article_pageviews for per-article pageviews.

Examples

# Basic call
enwiki_1_october_pageviews <- project_pageviews()

# Break it down to hourly
enwiki_hourly <- project_pageviews(granularity = "hourly", end = "2015100123")
Description

top_articles grabs data on the top articles for a project in a given time period, and for a particular platform.

Usage

top_articles(project = "en.wikipedia", platform = "all", start = as.Date("2015-10-01"), granularity = "daily", reformat = TRUE, ...)

Arguments

- project: the name of the project, structured as [language_code].[project] (see the default).
- platform: The platform the pageviews came from; one or more of "all", "desktop", "mobile-web" and "mobile-app". Set to "all" by default.
- start: The date the articles were "top" in. 2015 by default.
- granularity: the granularity of data to return; "daily" or "monthly", depending on whether top articles should reflect trends in day or month of the start date
- reformat: Whether to reformat the results as a data.frame or not. TRUE by default.
- ...: further arguments to pass to httr's GET.

See Also

- article_pageviews for per-article pageviews and project_pageviews for per-project pageviews.

Examples

# Basic example
enwiki_top_articles <- top_articles()

# Use a narrower platform
enwiki_mobile_top <- top_articles(platform = "mobile-web")
Index

article_pageviews, 2, 3–5

data.frame, 2, 4, 5
Date, 3

pageview_timestamps, 2, 3, 4
pageviews, 3
pageviews-package (pageviews), 3
POSIXlt, 3
project_pageviews, 2, 3, 4, 5

top_articles, 2, 4, 5