

Package ‘pageviews’

May 10, 2020

Title An API Client for Wikimedia Traffic Data

Version 0.5.0

Date 2020-05-10

Author Os Keyes [aut, cre], Jeremiah Lewis [ctb]

Maintainer Os 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

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-10 21:40:03 UTC

R topics documented:

article_pageviews	2
old_pageviews	3
pageviews	4
pageview_timestamps	4
project_pageviews	5
top_articles	6

article_pageviews	<i>Retrieve Pageview Data for an Article</i>
-------------------	--

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 = "2015100100",
  end = NULL,
  reformat = TRUE,
  granularity = "daily",
  ...
)
```

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 "automated". "all" 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 of data.
reformat	Whether to reformat the results as a data.frame or not. TRUE by default.
granularity	the granularity of data to return; "daily" or "monthly", depending on whether pageview data should reflect trends in days or months.
...	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")
```

old_pageviews	<i>Retrieve Legacy Pageview Counts</i>
---------------	--

Description

This retrieves per-project pageview counts from January 2008 to July 2016. These counts are calculated using the 'legacy' (read: old) model, which overcounts due to its inclusion of web-crawlers and similar automata.

Usage

```
old_pageviews(
  project = "en.wikipedia",
  platform = "all",
  granularity = "daily",
  start = "2013100100",
  end = "2015100100",
  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" or "mobile". Set to "all" by default.
granularity	the granularity of data to return; do you want hourly, daily or monthly 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 <code>granularity</code>).
reformat	Whether to reformat the results as a data.frame or not. TRUE by default.
...	further arguments to pass to htrr's GET.

See Also

[top_articles](#) for the top articles per project in a given date range, [project_pageviews](#) for per-project pageviews under the new definition, and [article_pageviews](#) for per-article pageviews.

Examples

```
# Basic call
enwiki_2013_2015_old <- old_pageviews()

# Break it down to hourly
old_enwiki_hourly <- old_pageviews(granularity = "hourly", end = "2013110100")
```

pageviews	<i>An API client for Wikimedia traffic data</i>
-----------	---

Description

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

pageview_timestamps	<i>Validate and convert time objects to function with pageviews functions</i>
---------------------	---

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 "automated". "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 <code>granularity</code>).
reformat	Whether to reformat the results as a data.frame or not. TRUE by default.
...	further arguments to pass to <code>httr</code> 's GET.

See Also

[old_pageviews](#), for 2008-2016 data, [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")
```

top_articles

Retrieve Data on Top Articles

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](#), [4](#), [6](#)

data.frame, [2](#), [3](#), [5](#), [6](#)

Date, [4](#)

old_pageviews, [3](#), [6](#)

pageview_timestamps, [2](#), [3](#), [4](#), [5](#)

pageviews, [4](#)

POSIXlt, [4](#)

project_pageviews, [2](#), [4](#), [5](#), [6](#)

top_articles, [2](#), [4](#), [6](#), [6](#)