

# Package ‘getwiki’

July 22, 2025

**Title** R Wrapper for Wikipedia Data

**Version** 0.9.0

**Description** A simple wrapper for 'Wikipedia' data. Specifically, this package looks to fill a gap in retrieving text data in a tidy format that can be used for Natural Language Processing.

**Imports** jsonlite

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Corydon Baylor [aut, cre]

**Maintainer** Corydon Baylor <cw Baylor1@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-05-12 08:20:05 UTC

## Contents

clean_wiki . . . . .	2
get_wiki . . . . .	2
random_wiki . . . . .	3
search_wiki . . . . .	3
trend_wiki . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

clean_wiki	<i>Clean Your Wiki</i>
------------	------------------------

---

**Description**

This function accepts a string and removes new line indentations and html tags with regex

**Usage**

```
clean_wiki(string)
```

**Arguments**

string	The string to be cleaned
--------	--------------------------

**Examples**

```
clean_wiki("<p>some text</p>")
```

---

get_wiki	<i>Get the Text of a Wikipedia Article</i>
----------	--

---

**Description**

Get the text of a wikipedia article by searching a title. For example, entering the search term "France" will return the text of the wikipedia page for France.

**Usage**

```
get_wiki(title, clean = TRUE)
```

**Arguments**

title	The title or (titles) of the Wikipedia page to be searched. If you would like to query multiple articles, put the titles in a character vector. The maximum number of titles that can be queried at one time is 50.
clean	Should getwiki remove html tags from the returned text?

**Value**

A single title will return the matched wikipedia article in a string. A vector of titles will return a dataframe with one column equal to the searched titles and one column equal to the matched article content

**Examples**

```
get_wiki("United States")  
get_wiki(c("United States", "France"))
```

---

random_wiki	<i>Get the Text of a Random Wikipedia Article</i>
-------------	---

---

**Description**

Get the text of a random wikipedia article

**Usage**

```
random_wiki(clean = TRUE)
```

**Arguments**

clean            Should getwiki remove html tags from the returned text?

**Value**

random\_wiki will return a single named character value whose value is the text of the wikipedia page

**Examples**

```
random_wiki()
```

---

search_wiki	<i>Search Wikipedia for Articles</i>
-------------	--------------------------------------

---

**Description**

Search for the top twenty wikipedia pages that match a given query. This function will return a dataframe with the names of the matched articles and the first paragraph of content.

**Usage**

```
search_wiki(search_term, clean = TRUE)
```

**Arguments**

search\_term    The search term you would like to use.  
clean           Should getwiki remove html tags from the returned text?

**Value**

search\_term will return a dataframe of the top twenty search results. The "title" column will be the title of the articles and the "content" column will be the first paragraph from those articles.

**Examples**

```
search_wiki("Belgrade")
```

---

```
trend_wiki
```

*Find the Page Views for an Article*

---

**Description**

Find the page views for an article for the past sixty days.

**Usage**

```
trend_wiki(title)
```

**Arguments**

`title`            The title of the Wikipedia article you would like trends for.

**Value**

`trend_wiki` will return a dataframe of the past sixty days of page views for the requested title

**Examples**

```
trend_wiki("Belgrade")
```

# Index

`clean_wiki`, 2

`get_wiki`, 2

`random_wiki`, 3

`search_wiki`, 3

`trend_wiki`, 4