Package 'wikiTools'

November 10, 2026

Type Package

Version 1.2.27

Date 2026-11-06

Title Tools for Wikidata and Wikipedia

Description A set of wrappers intended to check, read and download information from the Wikimedia sources. It is specifically created to work with names of celebrities, in which case their information and statistics can be downloaded. Additionally, it also builds links and snippets to use in combination with the function gallery() in netCoin package.

License GPL-3

Depends R (>= 3.5.0)

Encoding UTF-8

Imports curl, httr, jsonlite, collections, netCoin, evolMap

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Maintainer Modesto Escobar <modesto@usal.es>

RoxygenNote 7.3.2

Acknowledgments Grants CSO2013-49278-EXP, PGC2018-093755-B100, PDC2022-133355-100, PID2023-147358NB-100 funded by MICIU/AEI/10.13039/501100011033 and by "European Union NextGenerationEU/PRTR".

Author Modesto Escobar [aut, cph, cre] (ORCID: <https://orcid.org/0000-0003-2072-6071>),
Ángel Zazo [aut],
Carlos Prieto [aut] (ORCID: <https://orcid.org/0000-0001-8178-9768>),
David Barrios [aut],
Cristina Calvo [aut]

Repository CRAN

Date/Publication 2026-11-10 16:10:02 UTC

Contents

cc . . . 3
checkEntities . . . 4
checkTitles . . . 4
doChunks . . . 5
extractWiki . . . 5
filext . . . 6
getFiles . . . 7
getWikiData . . . 8
getWikiFiles . . . 9
getWikiInf . . . 10
httrGetJSON . . . 11
limitRequester . . . 12
m_Opensearch . . . 12
m_Pageviews . . . 13
m_reqMediaWiki . . . 14
m_XtoolsInfo . . . 16
nametoWikiFrame . . . 17
nametoWikiHtml . . . 18
nametoWikiURL . . . 19
normalizedTitle . . . 20
pop_up . . . 21
preName . . . 22
reqMediaWiki . . . 23
reqWDQS . . . 24
searchWiki . . . 24
selectLang . . . 25
urltoFrame . . . 26
urltoHtml . . . 27
user_agent . . . 28
validUrl . . . 28
v_AutoSuggest . . . 29
v_dfVIAF . . . 30
v_Extract . . . 31
v_GetRecord . . . 31
v_Search . . . 32
w_EntityInfo . . . 33
w_Exhibit . . . 35
w_Geoloc . . . 36
w_isInstanceOf . . . 37
w_isValid . . . 38
w_LabelDesc . . . 39
w_Property . . . 40
w_query . . . 42
w_SearchByAuthority . . . 42
w_SearchByIdentifiers . . . 44
w_SearchByInstanceof . . . 45

w_SearchByLabel 46
w_SearchByOccupation 48
w_Wikipedias 50

Index **52**

---

cc *Converts a text separated by commas into a character vector.*

Description

Converts a text separated by commas into a character vector.

Usage

```
cc(text, sep = ",")
```

Arguments

text	Text to be separated.
sep	A character of separation. It must be a blank. If it is another character, trailing blanks are suppressed.

Details

Returns inside the text are omitted.

Value

A vector of the split segments of the text.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## A text with three names separated with commas is converted into a vector of length 3.
cc("Pedro Almodovar, Diego Velazquez, Salvador Dali")
```

checkEntities *Check if all Wikidata entities in entity_list have valid values*

Description

Return a vector of entities with duplicates or void entities removed. A valid entity is a wikibase item (Qxxx, x is a digit) or a wikibase property (Pxxx).

Usage

```
checkEntities(entity_list)
```

Arguments

entity_list	A vector with the Wikidata entities.

Value

The list of entities or raise an error.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

checkTitles *checkTitles(titles) Check if titles are valid. Return TRUE is all titles are valid, else FALSE. See https://en.wikipedia.org/wiki/Wikipedia:Page_name#Technical_restrictions_and_limitations*

Description

checkTitles(titles) Check if titles are valid. Return TRUE is all titles are valid, else FALSE. See https://en.wikipedia.org/wiki/Wikipedia:Page_name#Technical_restrictions_and_limitations

Usage

```
checkTitles(titles)
```

Arguments

titles	A vector of titles to check.

doChunks *Execute a function in chunks.*

Description

Execute the function f(x,...) in chunks of chunk-size elements each. Wikidata and Wikimedia API have limits to execute a query. Wikidata has timeout limits, Wikimedia about the number of titles or pageIds. This function executes sequentially the function `f` over chunks of elements to prevent errors.

Usage

```
doChunks(f, x, chunksize, ...)
```

Arguments

`f`	The function to execute.
`x`	Vector of entities or titles/pageids.
`chunksize`	The number of elements in `x` to execute the function.
`...`	The `f` arguments.

Value

The results of execute `f` using all values of `x`

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

extractWiki *Extract the first paragraph of a Wikipedia article with a maximum of characters.*

Description

Extract the first paragraph of a Wikipedia article with a maximum of characters.

Usage

```
extractWiki(
  names,
  language = c("en", "es", "fr", "de", "it"),
  plain = FALSE,
  maximum = 1000
)
```

Arguments

names	A vector of names, whose entries have to be extracted.
language	A vector of Wikipedia's languages to look for. If the article is not found in the language of the first element, it search for the followings,.
plain	If TRUE, the results are delivered in plain format.
maximum	Number maximum of characters to be included when the paragraph is too large.

Value

a character vector with html formatted (or plain text) Wikipedia paragraphs.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## Not run:
## Obtaining information in English Wikidata
names <- c("William Shakespeare", "Pedro Almodovar")
info <- getWikiInf(names)
info$text <- extractWiki(info$label)

## End(Not run)
```

---

filext *Extract the extension of a file*

Description

Extract the extension of a file

Usage

```
filext(fn)
```

Arguments

fn	Character vector with the files whose extensions are to be extracted.

Details

This function extracts the extension of a vector of file names.

Value

A character vector of extension names.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## For a single item:
filext("Albert Einstein.jpg")
## You can do the same for a vector:
filext(c("Hillary Duff.png", "Britney Spears.jpg", "Avril Lavigne.tiff"))
```

---

getFiles *Downloads a list of files in a specified path of the computer, and return a vector of the no-found names (if any).*

Description

Downloads a list of files in a specified path of the computer, and return a vector of the no-found names (if any).

Usage

```
getFiles(lista, path = "./", ext = NULL)
```

Arguments

lista	A list or data frame of files' URLs to be download (See details).
path	Directory where to export the files.
ext	Select desired extension of the files. Default= NULL.

Details

This function allows download a file of files directly into your directory. This function needs a preexistent data frame of names and pictures' URL. It must be a list (or data.frame) with two values: "name" (specifying the names of the files) and "url" (containing the urls to the files to download).. All the errors are reported as outcomes (NULL= no errors). The files are donwload into your chosen directory.

Value

It returns a vector of errors, if any. All pictures are download into the selected directory (NULL= no errors).

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## Not run:

## In case you want to download a file directly from an URL:

# dta <- data.frame(name = "Data", url = "https://sociocav.usal.es/me/Stata/example.dta")
# getFiles(dta, path = "./")

##  You can can also combine this function with getWikiData (among others).
## In case you want to download a picture of a person:

# A <- data.frame(name= getWikiData("Rembrandt")$label, url=getWikiData("Rembrandt")$pics)
# getFiles(A, path = "./", ext = "png")

## Or the pics of multiple authors:

# B <- getWikiData(c("Monet", "Renoir", "Caillebotte"))
# data <- data.frame(name = B$label, url = B$pics)
# getFiles(data, path = "./", ext = NULL)

## End(Not run)
```

`getWikiData` *Create a data.frame with Wikidata of a vector of names.*

Description

Create a data.frame with Wikidata of a vector of names.

Usage

```
getWikiData(names, language = "en", csv = NULL)
```

Arguments

`names`	A vector consisting of one or more Wikidata's entry (i.e., topic or person).
`language`	The language of the Wikipedia page version. This should consist of an ISO language code (default = "en").
`csv`	A file name to save the results, in which case the only return is a message with the name of the saved file.

Value

A data frame with personal information of the names or a csv file with the information separated by semicolons.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## Obtaining information in English Wikidata
## Not run:
names <- c("William Shakespeare", "Pedro Almodovar")
info <- getWikiData(names)
## Obtaining information in Spanish Wikidata
d <- getWikiData(names, language="es")

## End(Not run)
```

`getWikiFiles` *Downloads a list of Wikipedia pages in a specified path of the computer, and return a vector of the no-found names (if any).*

Description

Downloads a list of Wikipedia pages in a specified path of the computer, and return a vector of the no-found names (if any).

Usage

```
getWikiFiles(X, language = c("es", "en", "fr"), directory = "./", maxtime = 0)
```

Arguments

`X`	A vector of Wikipedia's entry).
`language`	The language of the Wikipedia page version. This should consist of an ISO language code (default = "en").
`directory`	Directory where to export the files to.
`maxtime`	In case you want to apply a random waiting between consecutive searches.

Details

This function allows download a set of Wikipedia pages into a directory of the local computer. All the errors (not found pages) are reported as outcomes (NULL= no errors). The files are donwload into your chosen directory.

Value

It returns a vector of errors, if any. All pictures are download into the selected directory (NULL= no errors).

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## Not run:

## In case you want to download the Wikipage of a person:

# getWikiFiles("Rembrandt", dir = "./")

## Or the pics of multiple authors:

# B <- c("Monet", "Renoir", "Caillebotte")
# getWikiFiles(B, dir = "./", language="fr")

## End(Not run)
```

`getWikiInf` *Create a data.frame with Q's and descriptions of a vector of names.*

Description

Create a data.frame with Q's and descriptions of a vector of names.

Usage

```
getWikiInf(names, number = 1, language = "en")
```

Arguments

`names`	A vector consisting of one or more Wikidata's entry (i.e., topic or person).
`number`	Take the number occurrence in case there are several equal names in Wikidata.
`language`	The language of the Wikipedia page version. This should consist of an ISO language code (default = "en").

Value

A data frame with name, Q, label and description of the names.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## Obtaining information in English Wikidata
names <- c("William Shakespeare", "Pedro Almodovar")
information <- getWikiInf(names)

## Obtaining information in Spanish Wikidata
## Not run:
informacion <- getWikiInf(names, language="es")

## End(Not run)
```

`httrGetJSON` *httrGetJSON Retrieve responses in JSON format using httr::GET. It is a generic function to use for request these Wikimedia metrics API: https://wikimedia.org/api/rest_v1/ https://www.mediawiki.org/wiki/XTools/API/Page (xtools.wmflabs.org)*

Description

httrGetJSON Retrieve responses in JSON format using httr::GET. It is a generic function to use for request these Wikimedia metrics API: https://wikimedia.org/api/rest_v1/ https://www.mediawiki.org/wiki/XTools/API/Page (xtools.wmflabs.org)

Usage

```
httrGetJSON(url)
```

Arguments

`url`	The URL with the query to the API.

Value

A JSON response. Please check httr::stop_for_status(response)

Note

Used in m_Pageviews

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

limitRequester *Limits the rate at which a function will execute*

Description

Limits the rate at which a function will execute

Usage

```
limitRequester(f, n, period)
```

Arguments

f	The original function
n	Number of allowed events within a period
period	Length (in seconds) of measurement period

Value

If 'f' is a single function, then a new function with the same signature and (eventual) behavior as the original function, but rate limited. If 'f' is a named list of functions, then a new list of functions with the same names and signatures, but collectively bound by a shared rate limit. Used only for WikiData Query Service (WDQS).

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

m_Opensearch *Open search of a string*

Description

Search string in the content of the project page using OpenSearch. Only in namespace 0. Please, see https://www.mediawiki.org/wiki/API:Opensearch for further information.

Usage

```
m_Opensearch(
  string,
  project = "en.wikipedia.org",
  profile = "engine_autoselect",
  redirects = "resolve"
)
```

Arguments

string	String to search.
project	Wikimedia project, defaults "en.wikipedio.org".
profile	This parameter sets the search type: classic, engine_autoselect (default), fast-fuzzy, fuzzy, fuzzy-subphrases, normal, normal-subphrases, and strict.
redirects	If redirects='return', the page title is the normalized one (also the URL). If redirects='resolve", the page title is the normalized and resolved redirection is in effect (also the URL). Note that in both cases the API performs a NFC Unicode normalization on search string.

Value

A data-frame of page titles and URL returned. If error, return Null.

Note

Only for namespace 0. The function also obtains redirections for disambiguation pages.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
# Some search profiles:
df <- m_Opensearch(string='Duque de Alba', project='es.wikipedia.org',
                   profile="engine_autoselect", redirects="resolve")
df <- m_Opensearch(string='Duque de Alba', project='es.wikipedia.org', profile="strict")
df <- m_Opensearch(string='Duque de Alba', project='es.wikipedia.org', profile="fuzzy")
```

---

m_Pageviews *Get number of views of a Wikipedia article*

Description

Use the Wikimedia REST API (https://wikimedia.org/api/rest_v1/) to get the number of views one article has in a Wikimedia project in a date interval (see granularity). If redirect=TRUE, then get the number of views of all articles that redirects to the article which is the destiny of actual page.

Usage

```
m_Pageviews(
  article,
  start,
  end,
  project = "en.wikipedia.org",
  access = "all-access",
```

```
  agent = "user",
  granularity = "monthly",
  redirects = FALSE
)
```

Arguments

article	The title of the article to search. Only one article is allowed.
start, end	First and last day to include (format YYYYMMDD or YYYYMMDDHH)
project	The Wikimedia project, defaults en.wikipedia.org
access	Filter by access method: all-access (default), desktop, mobile-app, mobile-web
agent	Filter by agent type: all-agents, user (default), spider, automated
granularity	Time unit for the response data: daily, monthly (default)
redirects	Boolean to include the views of all redirections of the page (defaults: False). If redirects=TRUE then the "normalized" element of the returned vector contains the destiny of the redirection, and the "original" element contains the original title of the article. If a page is just a destiny of other pages, and you want to know the total number of views that page have (including views of redirections), it is also necessary set redirects=TRUE, otherwise only you have the views of that page.

Value

A vector with the number of visits by granularity.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
v <-  m_Pageviews(article="Cervantes", start="20230101", end="20230501",
                  project="es.wikipedia.org", granularity="monthly")
vv <- m_Pageviews(article="Cervantes", start="20230101", end="20230501",
                  project="es.wikipedia.org", granularity="monthly",
                  redirects=TRUE)
```

---

m_reqMediaWiki *Retrieve responses using the MediaWiki API.*

Description

Use the MediaWiki API to check Wikipedia pages titles, get redirections of Wikipedia pages, get image URL of Wikipedia pages or get URL of files in Wikipedia pages

Usage

```
m_reqMediaWiki(
  titles,
  mode = c("wikidataEntity", "redirects", "pagePrimaryImage", "pageFiles"),
  project = "en.wikipedia.org",
  redirects = TRUE,
  exclude_ext = "svg|webp|xcf"
)
```

Arguments

titles	A vector of page titles to search for.
mode	Select an action to perform: 'wikidataEntity' -> Use reqMediaWiki to check if page titles are in a Wikimedia project and returns the Wikidata entity for them. Automatically resolves redirects if parameter redirects = TRUE (default). If a page title exists in the Wikimedia project, the status column in the returned data-frame is set to 'OK'. If a page is a disambiguation page, that column is set to 'disambiguation', and if a title is not in the Wikimedia project, it is set to 'missing' and no Wikidata entity is returned; 'redirects' -> Obtains redirection of pages of the article titles in the Wikimedia project restricted to namespace 0. Returns a vector for each title, in each vector the first element is the page destiny, the rest are all pages that redirect to it. If a title is not in the Wikimedia project its list is NA; 'pagePrimaryImage' -> Return the URL of the image associated with the Wikipedia pages of the titles, if pages has one. Automatically resolves redirects, the "normalized" column of the returned data-frames contains the destiny page of the redirection. See https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bpageimages; 'pageFiles' -> Search for URL of files inserted in Wikipedia pages. Exclude extensions in exclude_ext. Note that the query API named this search as 'images', but all source files in the page are returned. The function only return URL that not end with extensions in exclude_ext parameter (case insensitive). Automatically resolves redirects, the "normalized" column of the returned data-frame contains the destiny page of the redirection. See https://en.wikipedia.org/w/api.php?action=help&modules
project	Wikimedia project, defaults "en.wikipedia.org"
redirects	If page redirects must be resolved. If redirects=TRUE (default) then the "normalized" column of the returned data-frames contains the destiny page title of the redirection. Only for mode=wikidataEntity.
exclude_ext	File extensions excluded in results. Only for mode=PageFiles. Default 'svg|webp|xcf'

Value

depends on the mode selected: 'wikidataEntity' Null if there is any error in response, else a data-frame with four columns: first, the original page title string, second, the normalized one, third, logical error=FALSE, if Wikidata entity exists for the page, or error=TRUE it does not, last, the Wikidata entity itself or a clarification of the error; 'redirects' A vector for each title, with all pages that are redirects to the first element; 'pagePrimaryImage' A data-frame with original titles, normalized ones, the status of the pages and the primary image of the page or NA if it does not

exist; 'pageFiles' A data-frame with original titles, the normalized ones, status for the page and the URL files of the Wikipedia pages, using use "|" to separate ones) or NA if files do not exits or are excluded.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
# Note that URLdecode("a%CC%8C") is
# the letter "a" with the combining caron
df <- m_reqMediaWiki(c('Max Planck', URLdecode("a%CC%8C"), 'Max', 'Cervante', 'humanist'),
                     mode='wikidataEntity', project='en.wikipedia.org')
a <- m_reqMediaWiki(c('Cervantes', 'Planck', 'Noexiste'), mode='redirects',
                     project='es.wikipedia.org')
i <- m_reqMediaWiki(c('Max Planck', URLdecode("a%CC%8C"), 'Max', 'Cervante', 'humanist'),
                     mode='pagePrimaryImage')
f <- m_reqMediaWiki(c('Max Planck', URLdecode("a%CC%8C"), 'Max', 'Cervante', 'humanist'),
                     mode='pageFiles', exclude_ext = "svg|webp|xcf")
```

---

m_XtoolsInfo *Gets various information from a Wikimedia page*

Description

Obtains information in JSON format about an article in the Wikimedia project or NULL on errors. Use the wmflabs API. The XTools Page API endpoints offer data related to a single page. See https://www.mediawiki.org/wiki/XTools/API/Page. The URL of the API starts with 'https://xtools.wmcloud.org/api/page/'

Usage

```
m_XtoolsInfo(
  article,
  infotype = c("articleinfo", "prose", "links"),
  project = "en.wikipedia.org",
  redirects = FALSE
)
```

Arguments

article	The title of the article to search. Only one article is allowed.
infotype	The type of information to request: articleinfo, prose, links. You also can type 'all' to retrieve all. Note that the API also offer theses options: top_editors, assessments, bot_data and automated_edits.
project	The Wikimedia project, defaults en.wikipedia.org.

redirects — If redirects==TRUE, then the information is obtained of the destiny of the page. In that case, then the "original" element of the returned list contains the original page, and the "page" element the destiny page. Also, if infotype==’links, the sum of the in-links of all redirections is assigned to links_in_count.

Value

A list with the information about the article.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
x <- m_XtoolsInfo(article="Cervantes", infotype="articleinfo", project="es.wikipedia.org")
xx <- m_XtoolsInfo(article="Cervantes", infotype="articleinfo", project="es.wikipedia.org",
                   redirects=TRUE)

y <- m_XtoolsInfo(article="Miguel de Cervantes", infotype="links", project="es.wikipedia.org")
yy <- m_XtoolsInfo(article="Cervantes", infotype="links", project="es.wikipedia.org",
                    redirects=TRUE)
z <- m_XtoolsInfo(article="Miguel de Cervantes", infotype="all", project="es.wikipedia.org")
zz <- m_XtoolsInfo(article="Cervantes", infotype="all", project="es.wikipedia.org",
                       redirects=TRUE)

## End(Not run)
```

---

nametoWikiFrame *Convert names into a Wikipedia’s iframe*

Description

Convert names into a Wikipedia’s iframe

Usage

```
nametoWikiFrame(name, language = "en")
```

Arguments

name — A vector consisting of one or more Wikipedia’s entry (i.e., topic or person).

language — The language of the Wikipedia page version. This should consist of an ISO language code (default = "en").

Details

This function adds the Wikipedia's iframe to a entry or name, i.e., "Max Weber" converts into "<iframe src=\"https://es.m.wikipedia.org/wiki/Max_Weber\" width=\"100...". It also manages different the languages of Wikipedia through the abbreviated two-letter language parameter, i.e., "en" = "english".

Value

A character vector of Wikipedia's iframes.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## When extracting a single item;
nametoWikiFrame("Computer", language = "en")

## When extracting two objetcs;
A <- c("Computer", "Operating system")
nametoWikiFrame(A)

## Same when three or more items;
B <- c("Socrates", "Plato", "Aristotle")
nametoWikiFrame(B)
```

---

nametoWikiHtml *Create the Wikipedia link of a name or entry.*

Description

Create the Wikipedia link of a name or entry.

Usage

```
nametoWikiHtml(name, language = "en")
```

Arguments

name	A vector consisting of one or more Wikipedia's entry (i.e., topic or person).
language	The language of the Wikipedia page version. This should consist of an ISO language code (default = "en").

Details

This function adds the Wikipedia's html link to a entry or name, i.e., "Max Weber" converts into "Max Weber". It also manages different the languages of Wikipedia through the abbreviated two-letter language parameter, i.e., "en" = "english".

Value

A character vector of names' links.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## When extracting a single item;
nametoWikiHtml("Computer", language = "en")

## When extracting two objetcs;
A <- c("Computer", "Operating system")
nametoWikiHtml(A)
B <- c("Socrates", "Plato","Aristotle" )
nametoWikiHtml(B)
```

---

nametoWikiURL *Create the Wikipedia URL of a name or entry.*

Description

Create the Wikipedia URL of a name or entry.

Usage

```
nametoWikiURL(name, language = "en")
```

Arguments

name	A vector consisting of one or more Wikipedia's entry (i.e., topic or person).
language	The language of the Wikipedia page version. This should consist of an ISO language code (default = "en").

Details

This function adds the Wikipedia URL to a entry or name, i.e., "Max Weber" converts into "https://es.wikipedia.org/wiki/Max_
It also manages different the languages of Wikipedia thru the abbreviated two-letter language parameter, i.e., "en" = "english".

Value

A character vector of names' URLs.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## When extracting a single item;
nametoWikiURL("Computer", language = "en")

## When extracting two objetcs;
A <- c("Computer", "Operating system")
nametoWikiURL(A)

## Same when three or more items;
B <- c("Socrates", "Plato" , "Aristotle")
nametoWikiURL(B)
```

---

normalizedTitle *Return the normalized and redirect title from the response*

Description

Return the normalized and the redirect title (also normalized), if any, from the query part of the JSON response of a MediaWiki search. The response of the MediaWiki API query (https://www.mediawiki.org/wiki/API:Que includes original page titles and possibly normalized and redirected titles, if the API needs to obtain them. For a original title, this function returns them, if any.

Usage

```
normalizedTitle(title, q)
```

Arguments

title	The title likely to be found in q.
q	The query part of the JSON response (j['query']) from a Mediawiki search. Note that this part contains some titles, so it is necessary to search the original "title" in that part.

Value

A vector with the normalized or redirected page title (target, also normalized) found for the title.

pop_up *Create a drop-down vignette for nodes from different items (for galleries).*

Description

Create a drop-down vignette for nodes from different items (for galleries).

Usage

```
pop_up(
  data,
  title = "name",
  title2 = NULL,
  info = TRUE,
  entity = "entity",
  links = c("wikidata", "wiki"),
  wikilangs = "en",
  dupBNE = FALSE
)
```

Arguments

data	Data frame which contains the data.
title	Column name which contains the first title of the vignette.
title2	Column name which contains the secondary title of the vignette.
info	Extract the first paragraph of a Wikipedia article.
entity	Column name which contains a vector of Wikidata entities.
links	Column names which contains the URLs for the vignette. 'wikidata' and 'wiki' by default, if this columns are missing, they will be generated through 'entity' argument.
wikilangs	List of languages to limit the search, using "|" as separator. Wikipedias page titles are returned in same order as languages in this parameter. If wikilangs=" the function returns Wikipedia page titles in any language, not sorted.
dupBNE	If this is true, it provides two links to the National Library of Spain (works and authors). Otherwise, only to the authority link.

Value

a character vector of html formatted vignettes attached to 'data' in a column named 'pop_up'.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## Not run:
library(netCoin)
data("sociologists")
sociologists$entity <- sub(".png","",sociologists$picture)
sociologists <- pop_up(sociologists, title="name",
  title2="birth_country", entity="entity")
plot(exhibit(sociologists, label="name", ntext="pop_up"))

## End(Not run)
```

`preName` *Reverse the order of the first and last names of every element of a vector.*

Description

Reverse the order of the first and last names of every element of a vector.

Usage

```
preName(X)
```

Arguments

`X`	A vector of names with format "name, prename".

Details

This function reverses the order of the first and last names of the items: i.e., "Weber, Max" turns into "Max Weber".

Value

Another vector with its elements changed.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## To reconvert a single name:
preName("Weber, Max")
## It is possible to work with several items, as in here:
A <- c("Weber, Max", "Descartes, Rene", "Locke, John")
preName(A)
```

reqMediaWiki *Uses httr package to retrieve responses using the MediaWiki API.*

Description

For MediaWiki requests only user_agent is necessary in the request headers. See https://www.mediawiki.org/wiki/API:Etique
The standard and default output format in MediaWiki is JSON. All other formats are discouraged.
The output format should always be specified using the request param "format" in the "query" request. See https://www.mediawiki.org/wiki/API:Data_formats#Output.

Usage

```
reqMediaWiki(
  query,
  project = "en.wikipedia.org",
  method = "GET",
  attempts = 2,
  debug = FALSE
)
```

Arguments

query	A list with de (key, values) pairs with the search. Note that if titles are included in the query, the MediaWiki API has a limit of 50 titles in each query. If number of titles is greater than this limit a error is raised.
project	The Wikimedia project to search. Default en.wikipedia.org.
method	The method used in the httr request. Default 'GET'. Note in "https://www.mediawiki.org/wiki/API:Etique "Whenever you're reading data from the web service API, you should try to use GET requests if possible, not POST, as the latter are not cacheable."
attempts	On ratelimit errors, the number of times the request is retried using a 60 seconds interval between retries. Default 2. If 0 no retries are done.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown.

Value

The response in JSON format, raise exception on errors.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

reqWDQS *Get responses from Wikidata Query Service*

Description

Retrieve responses from Wikidata Query Service (WDQS)

Usage

```
reqWDQS(sparql_query, format = "json", method = "GET")
```

Arguments

sparql_query	A string with the query in SPARQL language (SELECT query).
format	A string with the query response format, mandatory. See https://www.mediawiki.org/wiki/Wikidata_Quer Only 'json', 'xml' or 'csv' formats are allowed, default 'json'.
method	The method used in the httr request, GET or POST, mandatory. Default 'GET'. Use 'POST' method for long SELECT clauses.

Value

The response in the format selected. Please check httr::stop_for_status(response)

Note

For short queries GET method is better, POST for long ones. Only GET queries as cached.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

searchWiki *Find if there is a Wikipedia page of a name(s) in the selected language.*

Description

Find if there is a Wikipedia page of a name(s) in the selected language.

Usage

```
searchWiki(
  name,
  language = c("en", "es", "fr", "it", "de", "pt", "ca"),
  all = FALSE,
  maxtime = 0
)
```

Arguments

name	A vector consisting of one or more Wikipedia's entry (i.e., topic or person).
language	The language of the Wikipedia page version. This should consist of an ISO language code.
all	If all, all the languages are checked. If false, once a term is found, there is no search of others, so it's faster.
maxtime	In case you want to apply a random waiting between consecutive searches.

Details

This function checks any page or entry in order to find if it has a Wikipedia page in a given language. It manages the different the languages of Wikipedia thru the two-letters abbreviated language parameter, i.e, "en" = "english". It is possible to check multiple languages in order of preference; in this case, only the first available language will appear as TRUE.

Value

A Boolean data frame of TRUE or FALSE.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## When you want to check an entry in a single language:
searchWiki("Manuel Vilas", language = "es")

## When you want to check an entry in several languages:
## Not run:
searchWiki("Manuel Vilas", language = c( "en", "es", "fr", "it", "de", "pt", "ca"), all=TRUE)

## End(Not run)
## Not run:
A<-c("Manuel Vilas", "Julia Navarro", "Rosa Montero")
searchWiki(A, language = c("en", "es", "fr", "it", "de", "pt", "ca"), all=FALSE)

## End(Not run)
```

---

selectLang *Convert names of a wikiTools data frame to English or Spanish*

Description

Convert names of a wikiTools data frame to English or Spanish

Usage

```
selectLang(dbase, fields = names(dbase), language = "en")
```

Arguments

dbase	dataframe obtained by a wikiTools function .
fields	names of the dataframe to be translated (default: names of dbase).
language	default: "en". Also accept "es".

Value

the input dataframe with changed names

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca

---

urltoFrame *Convert an URL link to an HTML iframe.*

Description

Convert an URL link to an HTML iframe.

Usage

```
urltoFrame(url)
```

Arguments

url	Character vector of URLs.

Details

This function converts an available URL direction to the corresponding HTML iframe, i.e., "https://es.wikipedia.org/wiki/Soc
changes into "Socrates".

Value

A character vector of HTML iframe for the given urls.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## When you have a single URL:

urltoFrame("https://es.wikipedia.org/wiki/Socrates")

## It is possible to work with a vector of URL to obtain another vector of html frames:

A <- c("https://es.wikipedia.org/wiki/Socrates",
       "https://es.wikipedia.org/wiki/Plato",
       "https://es.wikipedia.org/wiki/Aristotle")
urltoHtml (A)
```

`urltoHtml` *Convert a Wikipedia URL to an HTML link*

Description

Convert a Wikipedia URL to an HTML link

Usage

```
urltoHtml(url, text = NULL)
```

Arguments

`url`	Character vector of URLs.
`text`	A vector with name of the correspondent title of the url (See details).

Details

This function converts an available URL direction to the corresponding HTML link, i.e., "https://es.wikipedia.org/wiki/Socrat
changes into "`<a href='https://es.wikipedia.org/wiki/Socrates' target='_blank'>Socrates</a>`".

Value

A character vector of HTML links for the given urls.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## When you have a single URL:

urltoHtml("https://es.wikipedia.org/wiki/Socrates", text = "Socrates")

## It is possible to work with several items:

A <- c("https://es.wikipedia.org/wiki/Socrates",
       "https://es.wikipedia.org/wiki/Plato",
       "https://es.wikipedia.org/wiki/Aristotle")
urltoHtml (A, text = c("Socrates", "Plato", "Aristotle"))

## And  you can also directly extract the info from nametoWikiURL():

urltoHtml(nametoWikiURL("Plato", "en"), "Plato" )
urltoHtml(nametoWikiURL(c("Plato", "Socrates", "Aristotle"), language="en"),
          c("Plato", "Socrates", "Aristotle"))
```

---

user_agent *See https://meta.wikimedia.org/wiki/User-Agent_policy https://www.mediawiki.org/wiki/API:Etiquette*

Description

See https://meta.wikimedia.org/wiki/User-Agent_policy https://www.mediawiki.org/wiki/API:Etiquette

Usage

```
user_agent
```

Format

An object of class `character` of length 1.

---

validUrl *Find if an URL link is valid.*

Description

Find if an URL link is valid.

Usage

```
validUrl(url, time = 2)
```

Arguments

url	A vector of URLs.
time	The timeout (in seconds) to be used for each connection. Default = 2.

Details

This function checks if a URL exists on the Internet.

Value

A boolean value of TRUE or FALSE.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
validUrl(url="https://es.wikipedia.org/wiki/Weber,_Max", time=2)
```

---

v_AutoSuggest *Suggests VIAF id from a name*

Description

Search the name of the author from the VIAF AutoSuggest API and returns information in JSON format of the records found. Note that only returns a maximum of 10 records. Note that those records are not VIAF cluster records. A VIAF record is considered a "cluster record," which is the result of combining records from many libraries around the world into a single record.

Usage

```
v_AutoSuggest(author)
```

Arguments

author	String to search. Please, see the structure of the author string to obtain better results: author: last name, first name[,] [([year_of_bird][-year_of_death])]

Value

A data-frame with four columns from the elements "term", "score", "nametype" and "viafid" of the Autosuggest API response.

See Also

https://developer.api.oclc.org/viaf-api#/Authority%20Cluster

Examples

```
v_AutoSuggest('Iranzo')
v_AutoSuggest('Esparza, Mar\uEDa')
# Four rows, only two viafid:
v_AutoSuggest('Escobar, Modesto')
```

---

v_dfVIAF *Flatten VIAF API Data into a Data Frame*

Description

Takes a parsed JSON response (as a list) from the Virtual International Authority File (VIAF) API and extracts key bibliographic and biographical fields. It flattens deeply nested and irregular list structures into a clean, single-row-per-record `data.frame`. Multiple values within a single field (such as titles, occupations, or sources) are collapsed and separated by a pipe (`|`).

Usage

```
v_dfVIAF(viaf, check_dateType = TRUE)
```

Arguments

`viaf`	A `list` object containing the parsed JSON data from the VIAF API. This can be a single record's list or a list containing multiple parsed records.
`check_dateType`	Logical. If `TRUE` (default), applies special formatting to dates based on their `dateType` attribute (e.g., adding "ca." for circa dates, or converting "flourished" dates into centuries).

Details

This function relies on several internal auxiliary functions (e.g., `v_titles`, `v_sources`, `v_wikipedias`) to safely extract specific fields, bypassing empty strings or unexpected data structures often returned by the API.

Value

A `data.frame` where each row represents a VIAF record. It contains the following columns: `VIAF_ID`, `Name_Type`, `Alternative_Names`, `Gender`, `Dates`, `Occupations`, `Countries`, `Unique_ISBNs`, `Titles`, `Sources`, `Coauthors`, and `Wikipedias`.

Examples

```
## Not run:
# Assuming you have already downloaded and parsed a JSON response:
# library(jsonlite)
# parsed_data <- fromJSON("http://viaf.org/viaf/96136385/viaf.json")

# Convert the parsed list into a clean data frame
```

```
df_viaf <- v_dfVIAF(viaf = parsed_data, check_dateType = TRUE)
head(df_viaf)

## End(Not run)
```

`v_Extract` *Gets information from a VIAF record*

Description

Returns information from the VIAF record. Note that the VIAF record musts be in JSON format.

Usage

```
v_Extract(viaf, info = "allinfo", source = NULL)
```

Arguments

`viaf`	VIAF cluster record (in JSON format).
`info`	is mandatory to select which information you want to retrieve. The options are 'titles', 'gender', 'dates', 'occupations', 'sources', 'sourceId' or 'wikipedias'.
`source`	the identifier of the source (LC, WKP, JPG, BNE...) Only if info=sourceId.

Value

depends on the info selected: 'titles' A list with titles; 'gender' The gender of the author o NULL if not exits in the record; 'dates' The bird year and death year in format byear:dyear; 'occupations' A data-frame with sources and occupations from each source or NULL if occupations do not exist in the record; 'sources' A data-frame with text and sources; 'sourceId' A data-frame with columns text and source, or NULL if the source does no exist in the viaf record; 'wikipedias' A vector with the URL of the Wikipedias.

`v_GetRecord` *Gets record clusters using viafID or other identifier in the VIAF cluster record.*

Description

Obtains the record cluster in JSON format. Other formats exists, but the more informative is 'application/json', so it is the only format used in this function. See https://developer.api.oclc.org/viaf-api

Usage

```
v_GetRecord(sid, source = "VIAF")
```

Arguments

sid	The identifier in the authority source.
source	The code of the authority source in VIAF (i.e, BNF, BNE, ISNI...).

Details

If source='VIAF' (default) the function uses the "Read by ID" interface of the API. If other authority source is used, then the function use the localAuthorityId/authoritySourceCode interface. See https://developer.api.oclc.org/viaf-api

Note that the returned record may be a VIAF cluster record or a redirect/scavenged record: the function returns the record as is.

Value

The VIAF record cluster in the JSON format.

Examples

```
## Not run:
a <- v_GetRecord('29550309')
b <- v_GetRecord('0000000059425637', source = 'ISNI')
c <- v_GetRecord('02853882X', source = 'SUDOC')
identical(a,b)
identical(b,c)

## End(Not run)
```

---

v_Search *Search the VIAF database using CQL queries*

Description

This function performs a search in the Virtual International Authority File (VIAF) database using Contextual Query Language (CQL). It handles pagination automatically to retrieve all matching records up to the API limits.

Usage

```
v_Search(
  CQL_Query,
 mode = c("default", "anyField", "allmainHeadingEl", "allNames", "allPersonalNames",
    "allTitle"),
  schema = c("VIAF", "brief")
)
```

Arguments

CQL_Query	A string containing the search terms or a full CQL query.
mode	A character string specifying the search field. Options are: • "default": Uses the raw CQL_Query. • "anyField": Searches in any available field. • "allmainHeadingEl": Searches in the main heading elements. • "allNames": Searches across all name variants. • "allPersonalNames": Limits search to personal names. • "allTitle": Searches in work titles.
schema	A character string specifying the record detail level. Use "VIAF" for full clusters or "brief" for a lightweight version.

Details

The function automatically cleans XML namespaces (e.g., "ns2:") from the resulting JSON to provide a cleaner list structure. It also ensures that viafID values are treated as characters to prevent precision loss.

Value

An object of class c("viaf", "list"). This is a list where each element is an individual VIAF record (also of class "viaf"). Returns NULL if no records are found or if a connection error occurs.

Examples

```
## Not run:
# Search for Miguel de Cervantes
results <- v_Search("Cervantes", mode = "allPersonalNames")

# Search for a specific title
works <- v_Search("Don Quixote", mode = "allTitle", schema = "brief")

## End(Not run)
```

---

w_EntityInfo *Get information about a Wikimedia entity (human or film)*

Description

Get labels, descriptions and some properties of the Wikidata entities in entity_list, for person or films. If person, the information returned is about labels, descriptions, birth and death dates and places, occupations, works, education sites, awards, identifiers in some databases, Wikipedia page titles (which can be limited to the languages in the wikilangs parameter, etc. If films, information is about title, directors, screenwriter, castmember, producers, etc.

Usage

```
w_EntityInfo(
  entity_list,
  mode = "default",
  langsorder = "",
  wikilangs = "",
  nlimit = MW_LIMIT,
  debug = FALSE
)
```

Arguments

entity_list	The Wikidata entities to search for properties (person or films.
mode	In "default" mode, the list of entities is expected to correspond to person, obtaining information related to person. If the mode is "film", information related to films will be requested. If the mode is "tiny" less properties are requested.
langsorder	Order of languages in which the information will be returned, separated with '|'. If no information is given in the first language, next is used. For label and description, English is used for language failback, if they are not in English, then information is returned in any else language. The language for label and description are also returned. If langsorder==", then no other information than labels or descriptions are returned in any language, only Wikidata entities, else, use the order in this parameter to retrieve information.
wikilangs	List of languages to limit the search of Wikipedia pages, using "|" as separator. Wikipedias pages are returned in same order as languages in this parameter. If wikilangs=" the function returns Wikipedia pages in any language, not sorted.
nlimit	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
debug	For debugging (info or query)

Value

A data-frame with the properties of the entity. Also index is set to entity_list.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
df <- w_EntityInfo(entity_list='Q134644', langsorder='es|en')
df <- w_EntityInfo(entity_list='Q134644', langsorder='es|en', wikilangs='es|en|fr')
df <- w_EntityInfo(c('Q270510', 'Q1675466', 'Q24871'), mode='film',
                   langsorder='es|en', wikilangs='es|en|fr')
# Search string 'abba' inlabel
w <- w_SearchByLabel('abba', mode='inlabel', langsorder = '', instanceof = 'Q5')
```

```
df <- w_EntityInfo(w$entity, langsorder='en', wikilangs='en|es|fr', debug='info')
# Search 3D films
w <- w_SearchByInstanceof(instanceof='Q229390', langsorder = 'en|es', debug = 'info')
df <- w_EntityInfo(w$entity, mode="film", langsorder='en', wikilangs='en', debug='info')

## End(Not run)
```

`w_Exhibit` *Extract the first paragraph of a Wikipedia article with a maximum of characters.*

Description

Extract the first paragraph of a Wikipedia article with a maximum of characters.

Usage

```
w_Exhibit(
  entities,
  mode = "default",
  langsorder = "en",
  wikilangs = langsorder,
  links = c("wikidata", "wiki", "BNE", "RAH", "ISNI"),
  map = c("none", "first", "last", "only"),
  info = FALSE,
  imgpath = NULL,
  nlimit = MW_LIMIT,
  debug = FALSE,
  ...
)
```

Arguments

`entities`	A vector or data.frame of entities, whose entries have to be extracted.
`mode`	type of data to be extracted, default=people.
`langsorder`	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. For label and description, English is used for language failback, if they are not in English, then information is returned in any else language. The language for label and description are also returned. If langsorder==", then no other information than labels or descriptions are returned in any language, only Wikidata entities, else, use the order in this parameter to retrieve information.
`wikilangs`	List of languages to limit the search of Wikipedia pages, using "|" as separator. Wikipedias pages are returned in same order as languages in this parameter. If wikilangs=" the function returns Wikipedia pages in any language, not sorted.

links	Vector of IDs for linking to its catalog. V.gr. c("Wikidata", "Wikipedia", "BNE", "RAH"). These four are those by default. Others possibles are c("BVMC", "VIAF", "LOC","ISNI", "ULAN", "OPENL", "MNCARS", "MNP", "SUDOC", "BRITISH", "BRITANNICA", "OXFORD", "WORLDCAT", "WEBCAT", "WEBGALL", "WIKIART")
map	A map should be created? This parameter can be "none", "only", "first" or "last".
info	Add the first paragraph of Wikipedia in the template.
imgpath	Name of the directory where there are image files.
nlimit	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
debug	For debugging (info or query).
...	Arguments for netCoin::exhibit(), evolMap::create_map() and evolMap::add_markers().

Value

An object of gallery_rd3 class.

Author(s)

Modesto Escobar, Department of Sociology and Communication, University of Salamanca. See https://sociocav.usal.es/blog/modesto-escobar/

Examples

```
## Not run:
## Obtaining information in English Wikidata
names <- c("William Shakespeare", "Pedro Almodovar")
info <- getWikiInf(names)
w_Exhibit(info$Q)

## End(Not run)
```

---

w_Geoloc *Get Latitude and Longitude coordinates, and Country of places*

Description

Get Latitude and Longitude coordinates of the Wikidata entities which are places. Also the countries they belong are returned.

Usage

```
w_Geoloc(entity_list, langsorder = "", nlimit = 1000, debug = FALSE)
```

Arguments

entity_list	A vector with de Wikidata entities (places).
langsorder	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. If langsorder==", then labels or descriptions are not returned.
nlimit	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown.

Value

A data-frame with 'entity', label, Latitude and Longitude, country and label of the country.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
w_Geoloc(c("Q57860", "Q90", "Q15695"), langsorder="")
w_Geoloc(c("Q57860", "Q90", "Q15695"), langsorder="se") # Note label of place for Q15695
w_Geoloc(c("Q57860", "Q90", "Q15695"), langsorder="se|fr")
df <- w_SearchByOccupation(Qoc='Q2306091') # aprox. 20000
l <- df$entity
# Get birth-place (P19)
p <- w_Property(l, Pproperty = 'P19', includeQ=TRUE, langsorder='es|en', debug='info')
# Filter entities that have places
places <- p[grepl("^Q\\d+$", p$P19), ]$P19
g <- w_Geoloc(places, langsorder='en|es', debug='info')

## End(Not run)
```

---

w_isInstanceOf *Check if a Wikidata entity is an instance of a class*

Description

Check using WDQS if the Wikidata entities in entity_list are instances of instanceof Wikidata entity class. For example, if instanceof="Q5", check if entities are instances of the Wikidata entity class Q5, i.e, are humans. Some entity classes are allowed, separately by 'l'; in this case, the OR operator is considered. If instanceof=" then no filter is applied: the function returns all Wikidata entities class of which each of the entities in the list are instances. Duplicated entities are deleted before search. Note that no labels or descriptions of the entities are returned. Please, use function w_LabelDesc for this.

Usage

```
w_isInstanceOf(entity_list, instanceof = "", nlimit = 50000, debug = FALSE)
```

Arguments

entity_list	A vector with the Wikidata entities.
instanceof	The Wikidata class to check, mandatory. Some entity classes separated by 'I' are allowed, in this case, the OR operator is considered.
nlimit	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown.

Value

A data-frame with three columns, first Wikidata entity, second all Wikidata class each instance is instance of them, last TRUE or FALSE if each entity is instance of the `instanceof` parameter, if this one is set.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
# aux: get a vector of entities (l).
df <- w_SearchByLabel(string='Iranzo', langsorder='es|en', mode='inlabel')
l <- df$entity

df <- w_isInstanceOf(entity_list=l, instanceof='Q5')
# Not TRUE
df[!df$instanceof_Q5,]

## End(Not run)
```

---

w_isValid *Check if Wikidata entities are valid*

Description

A entity is valid if it has a label or has a description. If one entity exists but is not valid, is possible that it has a redirection to other entity, in that case, the redirection is obtained. Other entities may have existed in the past, but have been deleted. The returned dataframe also includes the Wikidata class (another Wikidata entity) of which the searched entity are instances of. The data-frame no contains labels or descriptions about entities: the function `w_LabelDesc` can be used for valid entities. Duplicated entities are deleted before search. Index of the data-frame returned are also set to entity_list.

Usage

```
w_isValid(entity_list, nlimit = 50000, debug = FALSE)
```

Arguments

entity_list	A vector with de Wikidata entities.
nlimit	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown.

Value

A data-frame with four columns: entity, valid (TRUE or FALSE), instanceof and redirection (if the entity redirects to another Wikidata entity, the redirection column contains the last).

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
w_isValid(c("Q9021", "Q115637688", "Q105660123"))
# Large list
l  <- w_SearchByOccupation(Qoc='Q2306091')
l2 <- append(l$entity, c("Q115637688", "Q105660123"))  # Note: adding two new entities
v <- w_isValid(l2)
# Not valid
v[!v$valid, ]

## End(Not run)
```

---

w_LabelDesc *Return label and/or descriptions of Wikidata entities*

Description

Return label and/or descriptions of the entities in entity_list in language indicated in `langsorder`. Note that entities can be Wikidata entities (Qxxx) or Wikidata properties (Pxxx).

Usage

```
w_LabelDesc(
  entity_list,
  what = "LD",
  langsorder = "en",
  nlimit = 25000,
  debug = FALSE
)
```

Arguments

entity_list	A vector with de Wikidata entities.
what	Retrieve only Labels (L), only Descriptions (D) or both (LD).
langsorder	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. This parameter is mandatory, at least one language is required, default 'en'.
nlimit	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown.

Value

A data-frame with one column for the entities, and others for the language and the labels and/or descriptions. The index of the dataframe is also set to the entity list.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
w_LabelDesc(c("Q57860", "Q712609", "Q381800", "P569"), what='LD', langsorder = 'se|es|en')

## End(Not run)
```

---

w_Property *Get properties of Wikidata entities*

Description

Search the entities of the entity_list for property or properties. If searched properties can have more than one language, then the parameter langsorder set the order of language used. If parameter includeQ is TRUE, also is returned the Wikidata entities for the properties. The Wikidata class of which the entities are instances of are returned too. Duplicated entities are deleted before search. Index of the data-frame is also set to entity_list.

Usage

```
w_Property(
  entity_list,
  Pproperty,
  includeQ = FALSE,
  langsorder = "en",
  nlimit = 10000,
  debug = FALSE
)
```

Arguments

entity_list	A vector with de Wikidata entities.
Pproperty	Wikidata properties to search, separated with 'l', mandatory. For example, is Pproperty="P21", the results contain information of the sex of entities. If Pproperty="P21|P569" also searches for birthdate. If Pproperty='P21|P569|P214' also searches for VIAF identifier.
includeQ	If the value is TRUE the function returns the Wikidata entity (Qxxx) of the Pproperty. If also langsorder has language(s), the labels, if any, are returned too. Note that includeQ is only effective if Pproperty corresponds with a Wikidata entity, else the same values that label are returned.
langsorder	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. This parameter is mandatory if parameter includeQ if FALSE. If includeQ=TRUE and langsorder=" no labels are returned.
nlimit	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown.

Value

A data-frame with the entity, the entities of the properties and the labels in langsorder for them.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
w_Property(c("Q1252859", "Q712609", "Q381800"), Pproperty='P21|P569|P214', langsorder='en|es')
# Large list
df <- w_SearchByOccupation(Qoc='Q2306091') # ~ 20000
l <- df$entity
p <- w_Property(l, Pproperty='P21|P569|P214', langsorder='es|en', debug='info')
# Get birth-place (P19)
```

```
p <- w_Property(l, Pproperty='P19', langsorder='es|en', includeQ=TRUE, debug='info')

## End(Not run)
```

`w_query` *Response from Wikidata Query Service*

Description

Retrieve responses from Wikidata Query Service (WDQS). Uses a rate limiter if param limitRequester = TRUE.

Usage

```
w_query(sparql_query, format = "csv", method = "GET", limitRequester = FALSE)
```

Arguments

`sparql_query`	A string with the query in SPARQL language.
`format`	A string with the query response format. Mandatory. See https://www.mediawiki.org/wiki/Wikidata_Quer Only 'json', 'xml' or 'csv' formats are allowed, default 'csv'.
`method`	The method used in the httr request, GET or POST, mandatory. Default 'GET'.
`limitRequester`	If True, uses a rate limiter to limit the requests.

Value

The response in selected format or NULL on errors.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

---

`w_SearchByAuthority` *Get entities that have identifier in a database or authorities' catalog.*

Description

Get all Wikidata entities that have identifier in the database or authorities' catalog indicated in the parameter `Pauthority`. Returns the Wikidata entities. If parameter `langsorder=''`, then no labels or descriptions of the entities are returned, otherwise the function returns them in the language order indicated in `langsorder`. Filtering is possible if parameter `instanceof!=''`. If only the number of entities which have identifier in the database or authorities' catalog is needed, set debug='count'.

Usage

```
w_SearchByAuthority(
  Pauthority,
  langsorder = "",
  instanceof = "",
  nlimit = 10000,
  debug = FALSE
)
```

Arguments

Pauthority	Wikidata property identifier of the database or authorities' catalog. For example, if Pauthority = "P4439", all entities which have an identifier in the MNCARS (Museo Nacional Centro de Arte Reina Sofía) database are returnd. Following libraries abbreviation for the databases can be also used in the parameter 'Pauthority': library : VIAF, LC, BNE , ISNI, JPG, ULAN, BNF, GND, DNB, Pauthority: P214, P244, P950, P213, P245, P245, P268, P227, P227, library : SUDOC, NTA, J9U, ELEM, NUKAT, MNCARS, RAH Pauthority: P269, P1006, P8189, P1565, P1207, P4439, 13371
langsorder	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. If langsorder==", then labels or descriptions are not returned.
instanceof	Wikidata entity of which the entities searched for are an example or member of it (class). Optional. For example, if instanceof="Q5" the search are filtered to Wikidata entities of class Q5 (human). Some entity classes are allowed, separated with 'l'.
nlimit	If the number of entities in the database or authorities' catalog exceeds this number, then query are made in chunks. The value can increase if langorder=". Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown. If debug='count' the function only returns the number of entities with have identifier in that authority.

Value

A data-frame with columns: 'entity', 'entityLabel', 'entityDescription', 'instanceof', instanceofLabel' and the identifier in the "Pauthority" database. Index of the data-frame is also set to the list of entities found.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
# Example: Pauthority=P4439 (has identifier in the Museo Nacional Centro de
# Arte Reina Sof&iacute;a)
w_SearchByAuthority(Pauthority="P4439", debug='count')
mncars <- w_SearchByAuthority(Pauthority="P4439")
mncars <- w_SearchByAuthority(Pauthority="MNCARS", langsorder = 'es|en')
# Wikidata entities are not 'human' (Q5):
mncars[!grepl("\\bQ5\\b", mncars$instanceof), ]
# Wikidata entities are 'human' (Q5):
mncars <- w_SearchByAuthority(Pauthority="MNCARS", langsorder = 'es|en', instanceof='Q5')

## End(Not run)
```

---

w_SearchByIdentifiers *Search for entities that may match identifiers in a database or authorities' catalog.*

Description

The identifiers are in id_list. The database or authorities' catalog to which these identifiers belong must be provided in parameter `Pauthority`. If parameter langsorder=", then no labels or descriptions of the entities are returned, otherwise the function returns them in the language order indicated in `langsorder`. Duplicated entities are deleted before search. Index of the data-frame returned are also set to id_list.

Usage

```
w_SearchByIdentifiers(
  id_list,
  Pauthority,
  langsorder = "",
  nlimit = 3000,
  debug = FALSE
)
```

Arguments

`id_list`	List of identifiers.
`Pauthority`	Wikidata property identifier of the database or authorities' catalog. For example, if Pauthority = "P4439", then the function searches for entities that have the identifiers in the MNCARS (Museo Nacional Centro de Arte Reina Sofía) database. Following library abbreviations for the databases can be also used in the parameter 'Pauthority': library : VIAF, LC, BNE , ISNI, JPG, ULAN, BNF, GND, DNB, Pauthority: P214, P244, P950, P213, P245, P245, P268, P227, P1292 library : SUDOC, NTA, J9U, ELEM, NUKAT, MNCARS, RAH Pauthority: P269, P1006, P8189, P1565, P1207, P4439, P13371

langsorder	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. If langsorder=", then labels or descriptions are not returned.
nlimit	If the number of entities in the database or authorities' catalog exceeds this number, then query are made in chunks. The value can increase if langorder=". Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown. If debug='count' the function only returns the number of entities with have identifier in that authority.

Value

A data-frame with columns: 'entity', 'entityLabel', 'entityDescription', 'instanceof', instanceofLabel' and the identifier in the "Pauthority" database. Index of the data-frame is also set to the list of entities found.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
w_SearchByIdentifiers(c("4938246", "36092166", "40787112"), Pauthority='P214')
w_SearchByIdentifiers(c("4938246", "36092166", "40787112"), Pauthority='P214', langsorder='en|fr')

## End(Not run)
```

---

w_SearchByInstanceof *Get entities which are instance of a Wikidata entity*

Description

Get all Wikidata entities which are instance of one o more Wikidata entities like films, cities, etc. If parameter `langsorder=`", then no labels or descriptions of the entities are returned, otherwise the function returns them in the language order indicated in `langsorder`.

Usage

```
w_SearchByInstanceof(instanceof, langsorder = "", nlimit = 2500, debug = FALSE)
```

Arguments

instanceof	Wikidata entity of which the entities searched for are an example or member of it (class). For example, if instanceof="Q229390" return Wikidata entities of class Q229390 (3D films). More than one entities can be included in the instanceof parameter, with 'l' or '&' separator: • if 'l' (instanceof='Q229390|Q202866') then the OR operator is used. • if '&' (instanceof='Q229390|Q202866') then the AND operator is used. Note that 'l' and '&' cannot be present at the same time.
langsorder	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. If langsorder==", then labels or descriptions are not returned.
nlimit	If the number of entities in the database or authorities' catalog exceeds this number, then query are made in chunks. The value can increase if langorder=". Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown. If debug='count' the function only returns the number of entities.

Value

A data-frame. Index of the data-frame is also set to the list of entities found.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
w <- w_SearchByInstanceof('Q229390|Q25110269', langsorder = 'es|en')
w <- w_SearchByInstanceof('Q229390&Q25110269', langsorder = 'es|en')

## End(Not run)
```

---

w_SearchByLabel *Search Wikidata entities by string (usually labels)*

Description

Search Wikidata entities in label and altLabel ("Also known as") or in any part of the entity using different approaches.

Usage

```
w_SearchByLabel(
  string,
  mode = "inlabel",
  langs = "",
  langsorder = "",
  instanceof = "",
  Pproperty = "",
  debug = FALSE
)
```

Arguments

`string`	String (label or altLabel) to search. Note that single quotation mark must be escaped (string="O\'Donell"), otherwise an error will be raised.
`mode`	The mode to perform search. Default 'inlabel' mode. • 'exact' for an exact search in label or altLabel using case sensitive search and differentiate diacritics. Languages in the parameter `lang` are used, so this parameter is mandatory using this mode. • 'startswith' for entities which label or altLabel starts with the string, similar to a wildcard search "string*". The string is searched in label in the languages of `lang` parameter, but in any language in altLabel, so parameter `lang` is also mandatory in this mode. Diacritics and case are ignored in this mode. • 'cirrus' search words in any order in any part of the entity (which must be a string), not only in label or altLabel. Diacritics and case are ignored. It is a full text search using the ElasticSearch engine. Phrase search can be used if launched with double quotation marks, for example, string='"Antonio Saura"'. Also fuzzy search is possible, for example, string="algermon~1" or string="algernon~2". Also REGEX search can be used (but it is a very limited functionality) using this format: string="insource:/regex/i" (i: is for ignore case, optional). In this mode, parameter `langs` is ignored. • 'inlabel' is an special case of 'cirrus' search for matching whole words (in any order) in any position in label or altLabel. With this mode no fuzzy search can be used, but some languages can be set in the `lang` parameter. Modes 'inlabel' and 'cirrus' use the CirrusSearch of the Wikidata API. Please, for more examples, see https://www.mediawiki.org/wiki/Help:CirrusSearch and https://www.mediawiki.org/wiki/Help:Extension:WikibaseCirrusSearch
`langs`	Languages in which the information will be searched, using "|" as separator. In 'exact' or 'startswith' modes this parameter is mandatory, at least one language is required. In 'inlabel'mode, if the parameter `langs` is set, then the search is restricted to languages in this parameter, otherwise any language. In 'cirrus' mode this parameter is ignored.
`langsorder`	Order of languages in which the information will be returned, using "|" as separator. If `langsorder=`", no labels or descriptions will be returned, otherwise, they are returned in the order of languages in this parameter, if any.

instanceof	Wikidata entity of which the entities searched for are an example or member of it (class). For example, if instanceof='Q5' the search are filtered to Wikidata entities of class Q5 (human). Some entity classes are allowed, separated with '|'.
Pproperty	Wikidata properties to search, separated with '|', mandatory. For example, is Pproperty="P21", the results contain information of the sex of entities. If Pproperty="P21|P569" also searches for birthdate. If Pproperty='P21|P569|P214' also searches for VIAF identifier.
debug	For debugging purposes (default FALSE). If debug='query' the query launched is shown. If debug='count' the function only returns the number of entities with that occupation.

Value

A data-frame with 'entity', 'entityLabel', 'entityDescription', (including 'instance', 'instanceLabel', 'altLabel' if mode="startswith") and additionally the properties of Pproperty.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
df <- w_SearchByLabel(string='Iranzo', mode="exact", langs='es|en')
df <- w_SearchByLabel(string='Iranzo', mode="exact", langs='es|en',
                      langsorder='es|en', instanceof = 'Q5|Q101352')
## Search entities which label or altLabel starts with "string"
df <- w_SearchByLabel(string='Iranzo', mode='startswith', lang='en', langsorder='es|en')
## Search in any position in Label or AltLabel (diacritics and case are ignored)
df <- w_SearchByLabel(string='Iranzo', mode='inlabel', langsorder='es|en')
## Search in Chinese (Simplified) (language code: zh) in any part of entity:
df <- w_SearchByLabel(string='\u4F0A\u5170\u4f50', mode='cirrus', langsorder='es|zh|en')

## End(Not run)
```

---

w_SearchByOccupation *Get Wikidata entities with a certain occupation*

Description

Return the Wikidata entities which have the occupation indicated in Qoc, the Wikidata entity for that occupation. For example, if Qoc='Q2306091', returns the Wikidata entities which occupation is "Sociologist", among others. Also returns the Wikidata class of which the entities are instances of. If parameter langsorder='', then no labels or descriptions of the entities are returned, otherwise the function returns them in the language order indicated in langsorder. If wikilangs='' (if mode='wikipedias') then the Wikipedia pages are not filtered by language, else only Wikipedias of languages in this parameter are returned.

Usage

```
w_SearchByOccupation(
  Qoc,
  mode = c("default", "count", "wikipedias"),
  langsorder = "",
  wikilangs = "",
  nlimit = 10000,
  debug = FALSE
)
```

Arguments

Qoc	The Wikidata entity of the occupation. For example, Q2306091 for sociologist, Q2526255 for Film director, etc.
mode	The results you want to obtain: 'default' returns the Wikidata entities which have the occupation indicated; 'count' search in WDQS to know the number of Wikidata entities with that occupation); 'wikipedias' also the Wikipedia page of the entities are returned.
langsorder	Order of languages in which the information will be returned, separated with 'l'. If no information is given in the first language, next is used. If langsorder=", then labels or descriptions are not returned.
wikilangs	List of languages in Wikipedias to limit the search, using "|" as separator (only if mode='wikipedias'). Wikipedias page titles are returned in same order as languages in this parameter. If wikilangs=" the function returns Wikipedia page titles of entities in any language, not sorted.
nlimit	If the number of entities in that occupation exceeds this number, then query are made in chunks. The value can increase if langorder=". Please, reduce the default value if error is raised.
debug	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown. If debug='count' the function only returns the number of entities with that occupation.

Value

A data-frame with 'entity' and 'entityLabel', 'entityDescription', 'instanceof' and 'instanceofLabel' columns. Index of the data-frame is also set to the list of entities found.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
# "Q2306091" Qoc for Sociologist
w_SearchByOccupation(Qoc="Q2306091", mode='count')
q <- w_SearchByOccupation(Qoc="Q2306091", langsorder="")
```

```
q <- w_SearchByOccupation(Qoc="Q2306091", langsorder="en|es|fr")
q <- w_SearchByOccupation(Qoc="Q2306091", mode='wikipedias', debug='info')
q <- w_SearchByOccupation(Qoc="Q2306091", mode='wikipedias', wikilangs='en|es|fr', debug='info')

## End(Not run)
```

`w_Wikipedias` *Get Wikipedia pages of Wikidata entities*

Description

Get from Wikidata all Wikipedia page titles and URL of the Wikidata entities in entity_list. If parameter `wikilangs=''`, then returns all Wikipedia page titles, else only the languages in `wikilangs`. The returned dataframe also includes the Wikidata entity classes of which the searched entity is an instance. If set the parameter `instanceof`, then only returns the pages for Wikidata entities which are instances of the Wikidata class indicated in it. The data-frame doesn't return labels or descriptions about entities: the function `w_LabelDesc` can be used for this. Duplicated entities are deleted before search. Index of the data-frame returned are also set to entity_list.

Usage

```
w_Wikipedias(
  entity_list,
  wikilangs = "",
  instanceof = "",
  nlimit = 1500,
  debug = FALSE
)
```

Arguments

`entity_list`	A vector of Wikidata entities.
`wikilangs`	List of languages to limit the search, using "|" as separator. Wikipedias page titles are returned in same order as languages in this parameter. If wikilangs='' the function returns Wikipedia page titles in any language, not sorted.
`instanceof`	Wikidata entity class to limit the result to the instances of that class. For example, if instanceof='Q5', limit the results to "human".
`nlimit`	If the number of entities exceeds this number, chunked queries are done. This is the number of entities requested in each chunk. Please, reduce the default value if error is raised.
`debug`	For debugging purposes (default FALSE). If debug='info' information about chunked queries is shown. If debug='query' also the query launched is shown.

Value

A data-frame with five columns: entities, instanceof, npages, page titles and page URLs. Last three use "|" as separator. Index of data-frame is also set to the entity_list.

Author(s)

Angel Zazo, Department of Computer Science and Automatics, University of Salamanca

Examples

```
## Not run:
# aux: get a vector of entities (l).
df <- w_SearchByLabel(string='Napoleon', langsorder='en', mode='inlabel')
l <- df$entity  # aprox. 3600

w <- w_Wikipedias(entity_list=l, debug='info')
w <- w_Wikipedias(entity_list=l, wikilangs='es|en|fr', debug='info')
# Filter instanceof=Q5 (human):
w_Q5 <- w[grepl("\\bQ5\\b", w$instanceof), ]
w_Q5b <- w_Wikipedias(entity_list=l, wikilangs='es|en|fr', instanceof='Q5', debug='info')

## End(Not run)
```

Index

∗ **datasets**
    user_agent, 28

cc, 3
checkEntities, 4
checkTitles, 4

doChunks, 5

extractWiki, 5

filext, 6

getFiles, 7
getWikiData, 8
getWikiFiles, 9
getWikiInf, 10

httrGetJSON, 11

limitRequester, 12

m_Opensearch, 12
m_Pageviews, 13
m_reqMediaWiki, 14
m_XtoolsInfo, 16

nametoWikiFrame, 17
nametoWikiHtml, 18
nametoWikiURL, 19
normalizedTitle, 20

pop_up, 21
preName, 22

reqMediaWiki, 23
reqWDQS, 24

searchWiki, 24
selectLang, 25

urltoFrame, 26
urltoHtml, 27
user_agent, 28

v_AutoSuggest, 29
v_dfVIAF, 30
v_Extract, 31
v_GetRecord, 31
v_Search, 32
validUrl, 28

w_EntityInfo, 33
w_Exhibit, 35
w_Geoloc, 36
w_isInstanceOf, 37
w_isValid, 38
w_LabelDesc, 39
w_Property, 40
w_query, 42
w_SearchByAuthority, 42
w_SearchByIdentifiers, 44
w_SearchByInstanceof, 45
w_SearchByLabel, 46
w_SearchByOccupation, 48
w_Wikipedias, 50