Package 'WikipediR'

February 5, 2017

Type Package

Title A MediaWiki API Wrapper

Version 1.5.0

Date 2017-02-04

Author Oliver Keyes [aut, cre], Brock Tilbert [ctb]

Maintainer Oliver Keyes <ironholds@gmail.com>

Description A wrapper for the MediaWiki API, aimed particularly at the Wikimedia 'production' wikis, such as Wikipedia. It can be used to retrieve page text, information about users or the history of pages, and elements of the category tree.

License MIT + file LICENSE

Imports httr, jsonlite

Suggests testthat, knitr, WikidataR, pageviews

BugReports https://github.com/Ironholds/WikipediR/issues

URL https://github.com/Ironholds/WikipediR/

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-05 08:44:55

R topics documented:

`categories_in_page` *Retrieves categories associated with a page.*

Description

Retrieves categories associated with a page (or list of pages) on a MediaWiki instance

Usage

```
categories_in_page(language = NULL, project = NULL, domain = NULL, pages,
  properties = c("sortkey", "timestamp", "hidden"), limit = 50,
  show_hidden = FALSE, clean_response = FALSE, ...)
```

Arguments

`language`	The language code of the project you wish to query, if appropriate.
`project`	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with `language`.
`domain`	as an alternative to a `language` and `project` combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
`pages`	A vector of page titles, with or without spaces, that you want to retrieve categories for.
`properties`	The properties you want to retrieve about the categories. Options are "sortkey" (the key that sorts the way the page is stored in each category), "timestamp" (when the category was added to that page) and "hidden" (tags those categories in the returned list that are 'hidden' from readers).
`limit`	The maximum number of categories you want to retrieve for each page. Set to 50 by default.
`show_hidden`	Whether or not to include 'hidden' categories in the categories that are retrieved - these are usually associated with the maintenance of Wikipedia and its internal processes. Set to FALSE by default.
`clean_response`	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
`...`	further arguments to pass to httr's GET.

See Also

pages_in_category for pages in a specified category.

Examples

```
#Retrieve the categories for the "New Age" article on en.wiki
cats <- categories_in_page("en", "wikipedia", pages = "New Age")

#Retrieve the categories for the "New Age" article on rationalwiki.
rw_cats <- categories_in_page(domain = "rationalwiki.org", pages = "New Age")
```

---

pages_in_category *Retrieves a list of category members.*

Description

wiki_catpages retrieves a list of pages, subcategories, files or all of the above in a specified category (or series of specified categories)

Usage

```
pages_in_category(language = NULL, project = NULL, domain = NULL,
  categories, properties = c("title", "ids", "sortkey", "sortkeyprefix",
  "type", "timestamp"), type = c("page", "subcat", "file"),
  clean_response = FALSE, limit = 50, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
categories	The names of the categories you want to gather information for.
properties	The properties you want to gather for each member of the category. Options are "title" (the name of the member, including namespace), "id" (the unique numeric identifier of the member), "sortkey" (the hexadecimal key used to sort that member within the category), "sortkeyprefix" (the human-readable sort key), "type" (whether the member is a page, a subcategory or a file) and "timestamp" (when the member was added to the category)
type	The type of member you're interested in returning; options are any permutation of "page" (pages), "subcat" (subcategories) and "file" (files).
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.

limit	The maximum number of members to retrieve for each category. Set to 50 by default.
...	further arguments to pass to httr's GET().

warnings

Because of the way MediaWiki stores this data, both "the category you asked for doesn't exist" and "the category you asked for exists, but has no members" return in the same way.

See Also

categories_in_page for finding categories that a specified page is a member of.

Examples

```
#Retrieve the pages in the "New Age" category on en.wiki
cats <- pages_in_category("en", "wikipedia", categories = "New Age")

#Retrieve the pages in the "New Age" category on rationalwiki.
rw_cats <- pages_in_category(domain = "rationalwiki.org", categories = "New Age")
```

---

page_backlinks *Retrieve a page's backlinks*

Description

page_backlinks, when provided with a page title, retrieves backlinks to that page. Output can be filtered to specific namespaces.

Usage

```
page_backlinks(language = NULL, project = NULL, domain = NULL, page,
  limit = 50, direction = "ascending", namespaces = NULL,
  clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
page	the title of the page you want the backlinks of.
limit	the number of backlinks to return. Set to 50 (the maximum) by default.

direction	the direction to order the backlinks in, by linking page ID: "ascending" or "descending". Set to "ascending" by default.
namespaces	The namespaces to filter to. By default, backlinks from any namespace are retrieved: alternately, a numeric vector of accepted namespaces (which are described here) can be provided, and only backlinks from pages within those namespaces will be returned.
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

Warnings

as with pages_in_category, if the page you are linking to does not exist, an empty list will be returned, without any indication of an error.

Examples

```
#Backlink
all_bls <- page_backlinks("en","wikipedia", page = "Aaron Halfaker")

#Namespace-specific backlinks
mainspace_bls <- page_backlinks("en","wikipedia", page = "Aaron Halfaker", namespaces = 0)
```

---

page_content *Retrieves MediaWiki page content*

Description

wiki_page retrieves the DOM of a particular MediaWiki page, as a HTML blob inside a JSON object.

Usage

```
page_content(language = NULL, project = NULL, domain = NULL, page_name,
  page_id = NULL, as_wikitext = FALSE, clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
page_name	The title of the page you want to retrieve

page_id	the pageID of the page you want to retrieve. Set to NULL by default, and an alternative to page_name; if both are provided, page_id will be used.
as_wikitext	whether to retrieve the wikimarkup (TRUE) or the HTML (FALSE). Set to FALSE by default.
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

See Also

revision_diff for retrieving 'diffs' between revisions, revision_content for retrieving the text of specified revisions.

Examples

```
#Content from a Wikimedia project
wp_content <- page_content("en","wikipedia", page_name = "Aaron Halfaker")

#Content by ID
wp_content <- page_content("en", "wikipedia", page_id = 12)

#Content from a non-Wikimedia project
rw_content <- page_content(domain = "rationalwiki.org", page_name = "New Age")
```

---

page_external_links *Retrieve a page's links*

Description

page_external_links, when provided with a page title, retrieves external wikilinks from the current revision of that page.

Usage

```
page_external_links(language = NULL, project = NULL, domain = NULL, page,
  protocol = NULL, clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
page	the title of the page you want the links of.

protocol	limit links to those with certain link protocols. Options are listed in Special:ApiSandbox's elprotocol field.
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

Examples

```
#Links
external_links <- page_external_links("en","wikipedia", page = "Aaron Halfaker")

#Protocol-specific links
external_http_links <- page_external_links("en","wikipedia",
                                           page = "Aaron Halfaker", protocol = "http")
```

---

page_info *Retrieve information about a particular page*

Description

page_info, when provided with a page title, retrieves metadata about that page.

Usage

```
page_info(language = NULL, project = NULL, domain = NULL, page,
  properties = c("protection", "talkid", "url", "displaytitle"),
  clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
page	the title of the page you want the metadata of.
properties	the properties you'd like to retrieve. Some properties (the pageID, namespace, title, language, length and most recent revision ID, for example) are retrieved by default, whatever is passed to properties: properties that can be explicitly retrieved include the page's protection level ("protection"), the ID of the associated talk page, if applicable ("talkid"), the full, canonical URL ("url"), and the displayed page title ("displaytitle").
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

Examples

```
#Metadata
page_metadata <- page_info("en","wikipedia", page = "Aaron Halfaker")
```

`page_links` *Retrieve a page's links*

Description

page_links, when provided with a page title, retrieves internal wikilinks from the current revision of that page.

Usage

```
page_links(language = NULL, project = NULL, domain = NULL, page,
  limit = 50, direction = "ascending", namespaces = NULL,
  clean_response = FALSE, ...)
```

Arguments

`language`	The language code of the project you wish to query, if appropriate.
`project`	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with `language`.
`domain`	as an alternative to a `language` and `project` combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
`page`	the title of the page you want the links of.
`limit`	the number of links to retrieve. 50 by default; a maximum of 500 is set server-side.
`direction`	the direction to order the links in, by destination page ID: "ascending" or "descending". Set to "ascending" by default.
`namespaces`	The namespaces to filter to. By default, links to any namespace are retrieved: alternately, a numeric vector of accepted namespaces (which are described here) can be provided, and only backlinks from pages within those namespaces will be returned.
`clean_response`	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
`...`	further arguments to pass to httr's GET.

Examples

```
#Links
links <- page_links("en","wikipedia", page = "Aaron Halfaker")

#Namespace-specific links
mainspace_links <- page_links("en","wikipedia", page = "Aaron Halfaker", namespaces = 0)
```

query *base query function*

Description

not designed to be used by anyone except a third-party reuser package, such as WikidataR

Usage

```
query(url, out_class, clean_response = FALSE, query_param = list(), ...)
```

Arguments

url	a URL body
out_class	the class to set on the output object; used within WikidataR to indicate what response-cleaning method should be applied.
clean_response	whether to clean the response, using the method assigned by out_class, or not.
query_param	query parameters
...	further arguments to httr's GET.

random_page *Retrieve the page content of a random MediaWiki page*

Description

wiki_page retrieves the DOM of a particular MediaWiki page, as a HTML blob inside a JSON object.

Usage

```
random_page(language = NULL, project = NULL, domain = NULL,
  namespaces = NULL, as_wikitext = FALSE, limit = 1,
  clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.

namespaces	The namespaces to consider pages from. By default, pages from any namespace are considered; alternately, a numeric vector of accepted namespaces (which are described here) can be provided, and only pages within those namespaces will be considered.
as_wikitext	whether to retrieve the wikimarkup (TRUE) or the HTML (FALSE). Set to FALSE by default.
limit	the number of pages to return. 1 by default.
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

See Also

page_content for retrieving the content of a specific page, revision_diff for retrieving 'diffs' between revisions, revision_content for retrieving the text of specified revisions.

Examples

```
#A page from Wikipedia
wp_content <- random_page("en","wikipedia")

#A page from the mainspace on Wikipedia
wp_article_content <- random_page("en","wikipedia", namespaces = 0)
```

---

recent_changes *Retrieves entries from the RecentChanges feed*

Description

wiki_recentchanges retrieves a stream of entries from Special:RecentChanges, with a variety of associated metadata and filtering (of both entries *and* that metadata.

Usage

```
recent_changes(language = NULL, project = NULL, domain = NULL,
  properties = c("user", "userid", "comment", "parsedcomment", "flags",
  "timestamp", "title", "ids", "sizes", "redirect", "loginfo", "tags", "sha1"),
  type = c("edit", "external", "new", "log"), tag = NULL, dir = "newer",
  limit = 50, top = FALSE, clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.

domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
properties	Properties you're trying to retrieve about each entry, Options include "user" (the username of the person responsible for that entry), "userid" (the userID of said person), "comment" (the edit summary associated with the entry), "parsedcomment" (the same, but parsed, generating HTML from any wikitext in that comment), "flags" (whether the revision was 'minor' or not), "timestamp", "title" (the name of the page the entry affected), "ids" (the page id, along with the old and new revision IDs when applicable) "sizes" (the size, in uncompressed bytes, of the entry, and, in the case of revisions, the size of the edit it displaced), "tags" (any tags associated with the revision) and "loginfo" (applicable only to log entries, and consisting of log ID numbers, log types and actions, and so on) and "sha1" (the SHA-1 hash of the revision text).
type	The type of entry you want to retrieve; can be any permutation of "edit" (edits to existing pages), "external" (external actions that impact on the project - primarily wikidata changes), "new" (the creation of new pages) and "log" (log entries). By default, all of these entry types are included.
tag	Only return items with particular "tags", such as "mobile edit". NULL by default.
dir	Should it go from newest to oldest ("newer"), or oldest to newest ("older")? By default, set to "newer".
limit	The number of entries you'd like to return. By default, set to 50, which is also the maximum number per-request for logged-out users.
top	Should the request only return "top" entries - in other words, the most recent entry on a page? Set to FALSE by default.
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

`revision_content` *Retrieves MediaWiki revisions*

Description

Retrieves the content of a provided list of revisions from whichever MediaWiki instance you're querying. Returns as wikimarkup.

Usage

```
revision_content(language = NULL, project = NULL, domain = NULL,
  revisions, properties = c("content", "ids", "flags", "timestamp", "user",
  "userid", "size", "sha1", "contentmodel", "comment", "parsedcomment", "tags"),
  clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
revisions	The revision IDs of each desired revision.
properties	Properties you're trying to retrieve about that revision, should you want to; options include "ids" (the revision ID of the revision...which is pointless), "flags" (whether the revision was 'minor' or not), "timestamp" (the timestamp of the revision), "user" (the username of the person who made that revision), "userid" (the userID of the person who made the revision), "size" (the size, in uncompressed bytes, of the revision), "sha1" (the SHA-1 hash of the revision text), "contentmodel" (the content model of the page, usually "wikitext"), "comment" (the revision summary associated with the revision), "parsedcomment" (the same, but parsed, generating HTML from any wikitext in that comment), "tags" (any tags associated with the revision) and "flagged" (the revision's status under Flagged Revisions).
clean_response	whether to do some basic sanitising of the resulting data structure.
...	further arguments to pass to httr's GET.

See Also

revision_diff for diffs between revisions, and page_content for the content a specific page currently has.

Examples

```
#Revision content from a Wikimedia project
wp_content <- revision_content("en","wikipedia", revisions = 552373187)

#Revision content from a non-Wikimedia project
rw_content <- revision_content(domain = "rationalwiki.org", revisions = 88616)
```

---

revision_diff *Generates a "diff" between a pair of revisions*

Description

revision_diff generates a diff between two revisions in a MediaWiki page. This is provided as an XML-parsable blob inside the returned JSON object.

Usage

```
revision_diff(language = NULL, project = NULL, domain = NULL, revisions,
  properties = c("ids", "flags", "timestamp", "user", "userid", "size",
  "sha1", "contentmodel", "comment", "parsedcomment", "tags", "flagged"),
  direction, clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
revisions	The revision IDs of each "start" revision.
properties	Properties you're trying to retrieve about that revision, should you want to; options include "ids" (the revision ID of the revision...which is pointless), "flags" (whether the revision was 'minor' or not), "timestamp","user" (the username of the person who made that revision), "userid" (the userID of the person who made the revision), "size" (the size, in uncompressed bytes, of the revision), "sha1" (the SHA-1 hash of the revision text), "contentmodel" (the content model of the page, usually "wikitext"), "comment" (the revision summary associated with the revision), "parsedcomment" (the same, but parsed, generating HTML from any wikitext in that comment), "tags" (any tags associated with the revision) and "flagged" (the revision's status under Flagged Revisions).
direction	The direction you want the diff to go in from the revisionID you have provided. Options are "prev" (compare to the previous revision on that page), "next" (compare to the next revision on that page) and "cur" (compare to the current, extant version of the page).
clean_response	whether to do some basic sanitising of the resulting data structure.
...	further arguments to pass to httr's GET.

Warnings

MediaWiki's API is deliberately designed to restrict users' ability to make computing-intense requests - such as diff computation. As a result, the API only allows requests for one uncached diff in each request. If you ask for multiple diffs, some uncached and some cached, you will be provided with the cached diffs, one of the uncached diffs, and a warning.

If you're going to be asking for a lot of diffs, some of which may not be cached, it may be more sensible to retrieve the revisions themselves using revision_content and compute the diffs yourself.

See Also

page_content for retrieving the current content of a specific page, and revision_content for retrieving the text of specific revisions.

Examples

```
#Wikimedia diff
wp_diff <- revision_diff("en","wikipedia", revisions = 552373187, direction = "next")

#Non-Wikimedia diff
rw_diff <- revision_diff(domain = "rationalwiki.org", revisions = 88616, direction = "next")
```

`user_contributions` *Retrieve user contributions*

Description

Retrieves metadata associated with the most recent contributions by a specified user.

Usage

```
user_contributions(language = NULL, project = NULL, domain = NULL,
  username, properties = c("ids", "title", "timestamp", "comment",
  "parsedcomment", "size", "sizediff", "flags", "tags"), mainspace = FALSE,
  limit = 50, clean_response = FALSE, ...)
```

Arguments

`language`	The language code of the project you wish to query, if appropriate.
`project`	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with `language`.
`domain`	as an alternative to a `language` and `project` combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
`username`	The username of the user whose contributions you want to retrieve. Due to limitations at the API end, you can only retrieve edits for one user at a time.
`properties`	The metadata you want associated with each edit. Potential metadata includes "ids" (the revision ID of the revision, which can be passed into `revision_content`), "title" (the name of the page that was edited), "timestamp", "comment" (the edit summary associated with the revision), "parsedcomment" (the same, but parsed, generating HTML from any wikitext in that comment), "size" (the size, in uncompressed bytes, of the edit), "sizediff" (the size delta between this edit, and the last edit to the page), "flags" (whether the revision was 'minor' or not), and "tags" (any tags associated with the revision).
`mainspace`	A boolean flag; FALSE retrieves all of the most recent contributions, while TRUE limits the retrieved contributions to those in the 'mainspace' - in other words, edits to actual articles. Set to FALSE by default
`limit`	The number of edits to be retrieved. 50 is the maximum for logged-out API users, and putting in more than 50 will generate a warning.

clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

See Also

user_information for information about a specific user (or group of users), and recent_changes for non-user-specific recent actions.

Examples

```
#Retrieve the timestamps of a user's recent contributions to the English-language Wikipedia
contribs <- user_contributions("en", "wikipedia", username = "Ironholds",
                               properties = "timestamp")

#Retrieve the timestamps of a user's recent contributions to a non-Wikimedia wiki.
rw_contribs <- user_contributions(domain = "rationalwiki.org", username = "David Gerard",
                                  properties = "ids", limit = 1)
```

---

user_information *Retrieve user information*

Description

Retrieves information about a user, or set of users, from the MediaWiki API, including registration date, gender and editcount.

Usage

```
user_information(language = NULL, project = NULL, domain = NULL,
  user_names, properties = c("blockinfo", "groups", "implicitgroups",
  "rights", "editcount", "registration", "emailable", "gender"),
  clean_response = FALSE, ...)
```

Arguments

language	The language code of the project you wish to query, if appropriate.
project	The project you wish to query ("wikiquote"), if appropriate. Should be provided in conjunction with language.
domain	as an alternative to a language and project combination, you can also provide a domain ("rationalwiki.org") to the URL constructor, allowing for the querying of non-Wikimedia MediaWiki instances.
user_names	The username(s) of the users you want information on - this should be provided as a vector. There is a hard limit of 50 distinct users per query, set by MediaWiki's API; in the event that you go over this, a warning will be issued and the query will only be performed for the first 50 names in the vector.

properties	The user properties you're interested in. Applicable properties are "blockinfo" (details about the user's block, if they are currently blocked), "groups" (the user groups the user is a member of), "implicitgroups" (groups they are a member of through inheritance, as a result of membership in other groups), "rights" (what permissions their group membership grants them), "editcount" (how many non-deleted edits they have), "registration" (the date when they registered), "emailable" (whether they are contactable through Special:EmailUser) and "gender" (their provided gender).
clean_response	whether to do some basic sanitising of the resulting data structure. Set to FALSE by default.
...	further arguments to pass to httr's GET.

Warnings

There are a few caveats with the data provided by `user_information`, mostly stemming from historical inconsistencies and peculiarities in MediaWiki.

`groups` and `implicitgroups` gives you the user's permissions and group membership on the project you are querying, not their membership on all projects - while you can find out if "Ironholds" is not a sysop on, say, enwiki, that doesn't mean they aren't a sysop elsewhere - there is no universal, API-accessible user groups listing.

As an extension of the lack of centrality in Wikimedia's infrastructure, `registration` tells you the date their account was created on the wiki you are querying. If they initially registered on that wiki, this is accurate - if they registered on a different wiki, this instead reflects the date and time that they first visited the wiki you're querying while logged-in. For users registered before 2006, when registration logging was introduced, the `registration` value represents not when they first registered, but when their first edit was, since that was used as an estimator for existing accounts when the field was first populated.

See Also

`user_contributions` for retrieving recent contributions made by a particular user.

Examples

```
#Retrieving information from a Wikimedia project
user_info <- user_information("en", "wikipedia", user_names = "David Gerard",
                              properties = "registration")

#Non-Wikimedia projects
user_info <- user_information(domain = "rationalwiki.org", user_names = "David Gerard",
                              properties = "registration")
```

WikipediR *A client library for MediaWiki's API*

Description

This package provides functions for accessing the MediaWiki API, either for Wikimedia projects or any other MediaWiki instance. For more information, see the vignette.

See Also

The package vignette.

Index