Package ‘WikidataR’

September 22, 2017

Type Package

Title API Client Library for 'Wikidata'

Version 1.4.0

Date 2017-09-21

Author Oliver Keyes [aut, cre], Serena Signorelli [aut, cre], Christian Graul [ctb], Mikhail Popov [ctb]

Maintainer Oliver Keyes `<ironholds@gmail.com>`

Description An API client for the Wikidata <http://wikidata.org/> store of semantic data.

BugReports `https://github.com/Ironholds/WikidataR/issues`

URL `https://github.com/Ironholds/WikidataR/issues`

License MIT + file LICENSE

Imports httr, jsonlite, WikipediR (>= 1.4.0), utils

Suggests testthat, knitr, pageviews

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-09-22 05:43:08 UTC

R topics documented:

extract_claims 2
find_item 2
get_geo_box 3
get_geo_entity 4
get_item 5
get_random_item 6
print.find_item 7
print.find_property 7
print.wikidata 8
WikidataR 8

Index 9

extract_claims *Extract Claims from Returned Item Data*

Description

extract claim information from data returned using get_item.

Usage

```
extract_claims(items, claims)
```

Arguments

items	a list of one or more Wikidata items returned with get_item.
claims	a vector of claims (in the form "P321", "P12") to look for and extract.

Value

a list containing one sub-list for each entry in items, and (below that) the found data for each claim. In the event a claim cannot be found for an item, an NA will be returned instead.

Examples

```
# Get item data
adams_data <- get_item("42")

# Get claim data
claims <- extract_claims(adams_data, "P31")
```

find_item *Search for Wikidata items or properties that match a search term*

Description

find_item and find_property allow you to retrieve a set of Wikidata items or properties where the aliase or descriptions match a particular search term. As with other WikidataR code, custom print methods are available; use str to manipulate and see the underlying structure of the data.

Usage

```
find_item(search_term, language = "en", limit = 10, ...)

find_property(search_term, language = "en", limit = 10)
```

Arguments

`search_term`	a term to search for.
`language`	the language to return the labels and descriptions in; this should consist of an ISO language code. Set to "en" by default.
`limit`	the number of results to return; set to 10 by default.
`...`	further arguments to pass to httr's GET.

See Also

`get_random` for selecting a random item or property, or `get_item` for selecting a specific item or property.

Examples

```
#Check for entries relating to Douglas Adams in some way
adams_items <- find_item("Douglas Adams")

#Check for properties involving the peerage
peerage_props <- find_property("peerage")
```

`get_geo_box` *Get geographic entities based on a bounding box*

Description

get_geo_box retrieves all geographic entities in Wikidata that fall between a bounding box between two existing items with geographic attributes (usually cities).

Usage

```
get_geo_box(first_city_code, first_corner, second_city_code, second_corner,
  language = "en", ...)
```

Arguments

`first_city_code`	a Wikidata item, or series of items, to use for one corner of the bounding box.
`first_corner`	the direction of `first_city_code` relative to `city` (eg "NorthWest", "SouthEast").
`second_city_code`	a Wikidata item, or series of items, to use for one corner of the bounding box.
`second_corner`	the direction of `second_city_code` relative to `city` (eg "NorthWest", "SouthEast").
`language`	the two-letter language code to use for the name of the item. "en" by default.
`...`	further arguments to pass to httr's GET.

Value

a data.frame of 5 columns:

- item the Wikidata identifier of each object associated with `entity`.
- name the name of the item, if available, in the requested language. If it is not available, `NA` will be returned instead.
- latitude the latitude of `item`
- longitude the longitude of `item`
- entity the entity the item is associated with (necessary for multi-entity queries).

See Also

`get_geo_entity` for using an unrestricted search or simple radius, rather than a bounding box.

Examples

```
# Simple bounding box
bruges_box <- WikidataR:::get_geo_box("Q12988", "NorthEast", "Q184287", "SouthWest")

# Custom language
bruges_box_fr <- WikidataR:::get_geo_box("Q12988", "NorthEast", "Q184287", "SouthWest",
                                        language = "fr")
```

`get_geo_entity` *Retrieve geographic information from Wikidata*

Description

`get_geo_entity` retrieves the item ID, latitude and longitude of any object with geographic data associated with *another* object with geographic data (example: all the locations around/near/associated with a city).

Usage

```
get_geo_entity(entity, language = "en", radius = NULL, ...)
```

Arguments

`entity`	a Wikidata item (`Q...`) or series of items, to check for associated geo-tagged items.
`language`	the two-letter language code to use for the name of the item. "en" by default, because we're imperialist anglocentric westerners.
`radius`	optionally, a radius (in kilometers) around `entity` to restrict the search to.
`...`	further arguments to pass to httr's GET.

Value

a data.frame of 5 columns:

- item the Wikidata identifier of each object associated with `entity`.
- name the name of the item, if available, in the requested language. If it is not available, `NA` will be returned instead.
- latitude the latitude of `item`
- longitude the longitude of `item`
- entity the entity the item is associated with (necessary for multi-entity queries).

See Also

`get_geo_box` for using a bounding box rather than an unrestricted search or simple radius.

Examples

```
# All entities
sf_locations <- get_geo_entity("Q62")

# Entities with French, rather than English, names
sf_locations <- get_geo_entity("Q62", language = "fr")

# Entities within 1km
sf_close_locations <- get_geo_entity("Q62", radius = 1)

# Multiple entities
multi_entity <- get_geo_entity(entity = c("Q62", "Q64"))
```

`get_item` *Retrieve specific Wikidata items or properties*

Description

`get_item` and `get_property` allow you to retrieve the data associated with individual Wikidata items and properties, respectively. As with other `WikidataR` code, custom print methods are available; use `str` to manipulate and see the underlying structure of the data.

Usage

```
get_item(id, ...)

get_property(id, ...)
```

Arguments

`id` the ID number(s) of the item or property you're looking for. This can be in various formats; either a numeric value ("200"), the full name ("Q200") or even with an included namespace ("Property:P10") - the function will format it appropriately. This function is vectorised and will happily accept multiple IDs.

`...` further arguments to pass to httr's GET.

See Also

`get_random` for selecting a random item or property, or `find_item` for using search functionality to pull out item or property IDs where the descriptions or aliases match a particular search term.

Examples

```
#Retrieve a specific item
adams_metadata <- get_item("42")

#Retrieve a specific property
object_is_child <- get_property("P40")
```

`get_random_item` *Retrieve randomly-selected Wikidata items or properties*

Description

`get_random_item` and `get_random_property` allow you to retrieve the data associated with randomly-selected Wikidata items and properties, respectively. As with other `WikidataR` code, custom print methods are available; use `str` to manipulate and see the underlying structure of the data.

Usage

```
get_random_item(limit = 1, ...)

get_random_property(limit = 1, ...)
```

Arguments

`limit` how many random items to return. 1 by default, but can be higher.

`...` arguments to pass to httr's GET.

See Also

`get_item` for selecting a specific item or property, or `find_item` for using search functionality to pull out item or property IDs where the descriptions or aliases match a particular search term.

Examples

```
#Random item
random_item <- get_random_item()

#Random property
random_property <- get_random_property()
```

`print.find_item` *Print method for find_item*

Description

print found items.

Usage

```
## S3 method for class 'find_item'
print(x, ...)
```

Arguments

`x`	find_item object with search results
`...`	Arguments to be passed to methods

`print.find_property` *Print method for find_property*

Description

print found properties.

Usage

```
## S3 method for class 'find_property'
print(x, ...)
```

Arguments

`x`	find_property object with search results
`...`	Arguments to be passed to methods

print.wikidata *Print method for Wikidata objects*

Description

print found objects generally.

Usage

```
## S3 method for class 'wikidata'
print(x, ...)
```

Arguments

x	wikidata object from get_item, get_random_item, get_property or get_random_property
...	Arguments to be passed to methods

See Also

get_item, get_random_item, get_property or get_random_property

WikidataR *API client library for Wikidata*

Description

This package serves as an API client for https://www.wikidata.org. See the accompanying vignette for more details.

See Also

get_random for selecting a random item or property, get_item for a /specific/ item or property, or find_item for using search functionality to pull out item or property IDs where the descriptions or aliases match a particular search term.

Index

extract_claims, 2

find_item, 2, *6*, *8*
find_property *(*find_item*)*, 2

get_geo_box, 3, *5*
get_geo_entity, *4*, 4
get_item, *2, 3*, 5, *6*, *8*
get_property *(*get_item*)*, 5
get_random, *3*, *6*, *8*
get_random *(*get_random_item*)*, 6
get_random_item, 6
get_random_property *(*get_random_item*)*, 6

print.find_item, 7
print.find_property, 7
print.wikidata, 8

str, *2*, *5, 6*

WikidataR, 8
WikidataR-package *(*WikidataR*)*, 8