

Package ‘WikidataQueryServiceR’

July 21, 2025

Title API Client Library for 'Wikidata Query Service'

Version 1.0.0

Date 2020-06-16

Description An API client for the 'Wikidata Query Service'

<<https://query.wikidata.org/>>.

Depends R (>= 3.1.2)

Imports httr (>= 1.2.1), dplyr (>= 1.0.0), jsonlite (>= 1.2),
WikipediR (>= 1.5.0), ratelimitr (>= 0.4.1), purrr (>= 0.3.4),
readr (>= 1.3.1), rex (>= 1.2.0)

Suggests testthat (>= 2.3.0), lintr (>= 2.0.1)

URL <https://github.com/bearloga/WikidataQueryServiceR>

BugReports <https://github.com/bearloga/WikidataQueryServiceR/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Mikhail Popov [aut, cre] (@bearloga on Twitter),
Wikimedia Foundation [cph]

Maintainer Mikhail Popov <mikhail@wikimedia.org>

Repository CRAN

Date/Publication 2020-06-16 20:10:02 UTC

Contents

get_example	2
query_wikidata	3
WDQSR-deprecated	4

Index	5
--------------	----------

`get_example`*Get an example SPARQL query from Wikidata*

Description

Gets the specified example(s) from [SPARQL query service examples page](#) using [Wikidata's MediaWiki API](#).

Usage

```
get_example(example_name)
```

Arguments

`example_name` the names of the examples as they appear on [this page](#)

Details

If you are planning on extracting multiple examples, please provide all the names as a single vector for efficiency.

Value

The SPARQL query as a character vector.

See Also

[query_wikidata](#)

Examples

```
## Not run:
sparql_query <- extract_example(c("Cats", "Horses"))
query_wikidata(sparql_query)
# returns a named list with two data frames
# one called "Cats" and one called "Horses"

sparql_query <- extract_example("Largest cities with female mayor")
cat(sparql_query)
query_wikidata(sparql_query)

## End(Not run)
```

query_wikidata	<i>Send one or more SPARQL queries to WDQS</i>
----------------	--

Description

Makes a POST request to Wikidata Query Service SPARQL endpoint.

Usage

```
query_wikidata(sparql_query, format = c("simple", "smart"))
```

Arguments

sparql_query	SPARQL query (can be a vector of queries)
format	"simple" uses CSV and returns pure character data frame, while "smart" fetches JSON-formatted data and returns a data frame with datetime columns converted to POSIXct

Value

A tibble data frame

Query limits

There is a hard query deadline configured which is set to 60 seconds. There are also following limits:

- One client (user agent + IP) is allowed 60 seconds of processing time each 60 seconds
- One client is allowed 30 error queries per minute See [query limits section](#) in the WDQS user manual for more information.

See Also

[get_example](#)

Examples

```
sparql_query <- "SELECT
  ?softwareVersion ?publicationDate
WHERE {
  BIND(wd:Q206904 AS ?R)
  ?R p:P348 [
    ps:P348 ?softwareVersion;
    pq:P577 ?publicationDate
  ] .
}"
query_wikidata(sparql_query)
```

```
## Not run:  
query_wikidata(sparql_query, format = "smart")  
  
## End(Not run)
```

WDQSR-deprecated

Deprecated functions

Description

Why did I have to go and make things so deprecated?

Usage

```
scrape_example(example_name, ...)
```

Arguments

<code>example_name</code>	the names of the examples as they appear on this page
<code>...</code>	ignored (kept for backwards-compatibility)

Functions

- `scrape_example`: use [get_example](#) instead which employs [WikipediR::page_content](#)

Index

`get_example`, [2](#), [3](#), [4](#)

`query_wikidata`, [2](#), [3](#)

`scrape_example` (WDQSR-deprecated), [4](#)

WDQSR-deprecated, [4](#)

WikipediR::page_content, [4](#)