

Package ‘WikidataQueryServiceR’

April 29, 2017

Title API Client Library for 'Wikidata Query Service'

Version 0.1.1

Date 2017-04-28

Description An API client for the 'Wikidata Query Service'
<<https://query.wikidata.org/>>.

Depends R (>= 3.1.2)

Imports httr (>= 1.2.1), dplyr (>= 0.5.0), jsonlite (>= 1.2)

Suggests rvest (>= 0.3.2), urltools (>= 1.6.0)

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

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

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

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

Maintainer Mikhail Popov <mikhail@wikimedia.org>

Repository CRAN

Date/Publication 2017-04-28 22:26:11 UTC

R topics documented:

WDQS-package	2
query_wikidata	2
scrape_example	3

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

WDQS-package	<i>WikidataQueryServiceR: An R Wrapper For Wikidata Query Service API</i>
--------------	---

Description

This is an R wrapper for the [Wikidata Query Service](#) (WDQS) which provides a way for tools to query Wikidata via [SPARQL](#).

Details

[Wikidata Query Service](#) is maintained by [Wikimedia Foundation](#).

References

- [A beginner-friendly course for SPARQL](#)
- Building a SPARQL query: [Museums on Instagram](#)
- [SPARQL Query Examples](#) for WDQS
- [Using SPARQL to access Linked Open Data](#) by Matthew Lincoln
- Interesting or illustrative [SPARQL queries](#) for Wikidata
- [Wikidata 2016 SPARQL Workshop](#)
- [Wikidata SPARQL Query video tutorial](#) by Navino Evans
- [Learning SPARQL](#) by Bob DuCharme
- [WDQS User Manual](#)

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

Description

Makes a GET 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 POSIX1t
...	Additional parameters to supply to httr::GET()

Value

A data.frame

Examples

```
# R's versions and release dates:
sparql_query <- 'SELECT DISTINCT
  ?softwareVersion ?publicationDate
WHERE {
  BIND(wd:Q206904 AS ?R)
  ?R p:P348 [
    ps:P348 ?softwareVersion;
    pq:P577 ?publicationDate
  ] .
}'
query_wikidata(sparql_query)

## Not run:
# "smart" format converts all datetime columns to POSIXlt
query_wikidata(sparql_query, format = "smart")

## End(Not run)
```

scrape_example

Scrape an example SPARQL query from Wikidata

Description

Scrapes [SPARQL query service examples page](#) for specified example(s). Requires rvest and urltools packages.

Usage

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

Arguments

example_name	The names of the examples as they appear on this page
...	Additional httr configurations passed to rvest

Details

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

Value

The SPARQL query as a character vector.

Examples

```
## Not run:
sparql_query <- scrape_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 <- scrape_example("Largest cities with female mayor")
cat(sparql_query)
query_wikidata(sparql_query)

## End(Not run)
```

Index

`httr::GET()`, [2](#)

`query_wikidata`, [2](#)

`scrape_example`, [3](#)

WDQS (WDQS-package), [2](#)

WDQS-package, [2](#)