

Package ‘WikidataQueryServiceR’

January 17, 2017

Title API Client Library for 'Wikidata Query Service'

Version 0.1.0

Date 2017-01-17

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 5.0.1

NeedsCompilation no

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

Maintainer Mikhail Popov <mikhail@wikimedia.org>

Repository CRAN

Date/Publication 2017-01-17 23:12:16

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's [Discovery](#) Department.

References

- [WDQS User Manual](#)
- [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](#)

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 POSIXlt
...	Additional parameters to supply to GET

Value

A data.frame

Examples

```

# Cats on Wikidata:
sparql_query <- 'SELECT ?item ?itemLabel
WHERE
{
  ?item wdt:P31 wd:Q146 .
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
}
LIMIT 10'
query_wikidata(sparql_query)

## Not run:
sparql_query <- "#Recent Events
SELECT ?event ?eventLabel ?date
WHERE
{
  # find events
  ?event wdt:P31/wdt:P279* wd:Q1190554.
  # with a point in time or start date
  OPTIONAL { ?event wdt:P585 ?date. }
  OPTIONAL { ?event wdt:P580 ?date. }
  # but at least one of those
  FILTER(BOUND(?date) && DATATYPE(?date) = xsd:dateTime).
  # not in the future, and not more than 31 days ago
  BIND(NOW() - ?date AS ?distance).
  FILTER(0 <= ?distance && ?distance < 31).
  # and get a label as well
  OPTIONAL {
    ?event rdfs:label ?eventLabel.
    FILTER(LANG(?eventLabel) = \"en\").
  }
}
# limit to 10 results so we don't timeout
LIMIT 10"

# "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 http 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

GET, [2](#)

query_wikidata, [2](#)

scrape_example, [3](#)

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

WDQS-package, [2](#)