Get RDF data cube vocabulary as turtle file from qb: (http://purl.org/linked-data/cube#)

mja@statgroup.dk

2016-06-27

Contents

Get RDF data cube vocabulary as turtle file from qb: (http://purl.org/linked-data/cube#).

This script retrieves the RDF data cube vocabulary and stores it as a .ttl in the packages.

Setup

```
library(rrdfancillary)
devtools::load_all(pkg="../..")
```

```
## Loading rrdfqb
```

Get and store RDF data vocabulary

```
library(RCurl)
library(devtools)
qbURL<-"https://raw.githubusercontent.com/UKGovLD/publishing-statistical-data/master/specs/src/main/voca
if (! url.exists(qbURL) ) {
  stop(paste0("Can not access URL ",qbURL))
}
cube.vocabulary.ttl <- getURL(qbURL)
savefile <- file.path(system.file("extdata/cube-vocabulary-rdf", package="rrdfqb"), "cube.ttl" )
writeLines( cube.vocabulary.ttl, savefile)
cat("written to ", normalizePath(savefile), "\n" )
```

```
## written to  /home/ma/projects/rrdfqbcrnd0/rrdfqb/inst/extdata/cube-vocabulary-rdf/cube.ttl
```

SPARQL for getting the cube vocabulary using FROM in the SPARQL query

The following is experimental and not needed for the present package. See below for explanation.

```
rdf.data.cube.URLstem<- qbURL
SPARQLscript<- paste(
    "CONSTRUCT { ?s ?p ?o }",
    paste0( "FROM ", "<", rdf.data.cube.URLstem, ">", collapse="\n" ),
    "WHERE { ?s ?p ?o }",
    sep="\n", collapse="\n"
)
cat(SPARQLscript,"\n")
```

CONSTRUCT { ?s ?p ?o } FROM https://raw.githubusercontent.com/UKGovLD/publishing-statistical-data/master/specs/src/main/vocab/cube.ttl WHERE { ?s ?p ?o }

```
SPARQLscriptfn<- file.path(tempdir(),"get-cube.rq")
writeLines( SPARQLscript, con=SPARQLscriptfn )
cat("SPARQL script stored in  ", normalizePath(SPARQLscriptfn), "\n")
```

SPARQL script stored in /tmp/RtmpYuISnz/get-cube.rq

Using the script with R

The R-code below does not work with `rrdf`, presumably because the FROM clause is not evaluated.

```
store <- new.rdf()
summarize.rdf(store)
results.rdf <- construct.rdf(store, SPARQLscript )
summarize.rdf(results.rdf)

dump.rq<- paste(
    "SELECT  ?s ?p ?o ",
    "WHERE { ?s ?p ?o }",
    sep="\n", collapse="\n"
)
cat(dump.rq,"\n")
cube.triples<- data.frame(sparql.rdf(results.rdf, dump.rq ), stringsAsFactors=FALSE)
knitr::kable(cube.triples)
```

However, it works using a local Fuseki instance. The code below assumes a local Fuseki instance is running at `http://localhost:3030/arm/query`. Change to follow your setup.

```
endpoint<- "http://localhost:3030/arm/query"
results.fuseki <- construct.remote(endpoint, SPARQLscript )
summarize.rdf(results.fuseki)
```