Package: rtematres (via r-universe)

October 31, 2024

Title Exploit vocabularies on tematres server.

Description Exploit vocabularies on tematres server and annotate data frames in R.

Version 0.3

Author Claas-Thido Pfaff <claas-thido.pfaff@idiv-biodiversity.de>

Maintainer Claas-Thido Pfaff <claas-thido.pfaff@idiv-biodiversity.de>

URL https://github.com/cpfaff/rtematres

BugReports https://github.com/cpfaff/rtematres/issues

VignetteBuilder knitr

Suggests knitr, devtools, testthat

Imports XML, RCurl, plyr, gdata

Date 2013-08-13

License GPL-3

Repository https://cpfaff.r-universe.dev

RemoteUrl https://github.com/cpfaff/rtematres

RemoteRef HEAD

RemoteSha eadcc94ad633aa88f6b570c02922ca35919b4f90

Contents

`annotate.dataframe` *This is the annotation features provided by rtematres*

Description

You can semantically annotate data frames base on a tematres thesaurus.

Usage

```
annotate.dataframe(input)
```

Arguments

`input`	to be annotated takes a data frame

Value

a list wih annotation content

`annotate.dataframe.clean`
You can semantically annotate data frames base on a tematres thesaurus wih cleaning the dataset

---

Description

You can semantically annotate data frames base on a tematres thesaurus wih cleaning the dataset

Usage

```
annotate.dataframe.clean(input)
```

Arguments

`input`	to be annotated takes a data frame

Value

a list wih annotation content

rtematres *A convenient wrapper to all tasks of the base api.*

Description

As some of the task of the base api only take ids the wrapper does a conversion from a term to the id to communicate with the server. So you can use terms in all taks with this function.

rtematres A package to exploit controlled vocabularies from tematres servers

Usage

```
rtematres(task, verbose = F, term)
```

Arguments

task	The api task you like to execute.
verbose	Either true of false and determines the ammount of info that is returned by a query.
term	Is the term(s) you like to execute the task for.

Value

The function returns either a dataframe for information or a list of keywords and ids

Examples

```
## Not run:
    rtematres(task = "fetchVocabularyData")
    rtematres(task = "fetchTopTerms")
    rtematres(task = "fetchCode", term = "tree")
    rtematres(task = "search", term = "measurement")
    rtematres(task = "fetch", term = "measurement")
    rtematres(task = "searchNotes", term = "measurement")
    rtematres(task = "suggest", term = "measurement")
    rtematres(task = "suggestDetails", term = "measurement")
    rtematres(task = "fetchSimilar", term = "t")
    rtematres(task = "letter", term = "t")
    rtematres(task = "fetchAlt", term = "tree" )
    rtematres(task = "fetchTerm", term = "tree")
    rtematres(task = "fetchTerms", term = c("Context", "tree") )
    rtematres(task = "fetchDown", term = "Context")
    rtematres(task = "fetchUp", term = "measurement")
    rtematres(task = "fetchRelated", term = "tree")
    rtematres(task = "fetchRelatedTerms", term = c("Context", "tree"))
    rtematres(task = "fetchNotes", term = "Context")
    rtematres(task = "fetchDirectTerms", term = "carbon")
    rtematres(task = "fetchURI", term = "carbon")
    rtematres(task = "fetchTargetTerms", term = "carbon")
```

```
    rtematres(task = "fetchSourceTerms", term = "Context")
    rtematres(task = "fetchLast")

## End(Not run)
```

`rtematres.api` *Access to basic tematres server api*

Description

Features the tasks of the tematres server api. With no sugar added. They are the basic building blocks for more convenient user functions.

Usage

```
rtematres.api(task = "availableTasks", argument)
```

Arguments

`task`	The api task you like to perform. Use the the task "availableTasks" to get an overview about the base api. It returns a data frame with descriptions and the arguments for the tasks.
`argument`	Is the argument for the api task. You find the information about the type of arguments when you call the task "availableTasks". It depends on the task if the argument is numeric or a character.

Value

The function returns either a dataframe for "availableTasks" or a list of information elements for a certain task.

Examples

```
## Not run:
    rtematres.api(task = "availableTasks")
    rtematres.api(task = "fetchVocabularyData")
    rtematres.api(task = "fetchTopTerms")
    rtematres.api(task = "search", argument = "measurement")
    rtematres.api(task = "fetch", argument = "measurement")
    rtematres.api(task = "searchNotes", argument = "measurement")
    rtematres.api(task = "suggest", argument = "measurement")
    rtematres.api(task = "suggestDetails", argument = "measurement")
    rtematres.api(task = "fetchSimilar", argument = "tre")
    rtematres.api(task = "letter", argument = "t")
    rtematres.api(task = "fetchTerm", argument = 12)
    rtematres.api(task = "fetchDown", argument = 4 )
    rtematres.api(task = "fetchUp", argument = 4)
    rtematres.api(task = "fetchRelated", argument = 4)
    rtematres.api(task = "fetchAlt", argument = 12 )
```

```
    rtematres.api(task = "fetchCode", argument = "tree")
    rtematres.api(task = "fetchNotes", argument = 5 )
    rtematres.api(task = "fetchDirectTerms", argument = 12)
    rtematres.api(task = "fetchURI", argument = 12)
    rtematres.api(task = "fetchTargetTerms", argument = 12 )
    rtematres.api(task = "fetchSourceTerm", argument = "measurement")
    rtematres.api(task = "fetchTerms", argument = '12,13' )
    rtematres.api(task = "fetchRelatedTerms", argument = '12,13' )
    rtematres.api(task = "fetchLast")

## End(Not run)
```

`rtematres.common` *Find common concept for categorial columns*

Description

Search the thesaurus for concepts and extract their hirarchy. Then return the common concept if any.

Usage

```
rtematres.common(input)
```

Arguments

`input`	A categorial vector

Value

The function returns the common concept

Examples

```
## Not run:
 rtematres.common(c("carbon", "nitrogen", "organic carbon"))

## End(Not run)
```

rtematres.define *Convenient functions for various tasks*

Description

Define a concepts

Usage

```
rtematres.define(term)
```

Arguments

term	The concept you are looking for

Details

The function retrieves the definition of a term. This works of course only given the case it has been described in the vocuabulary you query.

Value

It returns a text string describing the concept of interest

rtematres.hierarchy *Locate concepts in hierarchy*

Description

Search the thesaurus for concepts and extract their hirarchy. This envolves higher order and lower order terms.

Usage

```
rtematres.hierarchy(term)
```

Arguments

term	The concept you are looking for

Value

The function returns a character vector of concepts

Examples

```
## Not run:
 rtematres.hierarchy("carbon")

## End(Not run)
```

---

rtematres.illuminate *Summarize information for a concept*

Description

Summarizes information for a concept in one convenient function call. Currently it provides the definition, upstream and down stream concepts for the term you querry for and related concepts

Usage

```
rtematres.illuminate(input)
```

Arguments

input	A concept name as string

Value

The function returns a list containing the information

Examples

```
## Not run:
    rtematres.illuminate(input = "carbon")

## End(Not run)
```

---

rtematres.options *Set or query otions related to the rtematresdata R package.*

Description

This function is used to query and set the options used by the rtematres package. For example you can set the URLs to your tematres server and the API.

Usage

```
rtematres.options(...)
```

Arguments

...	similar to options. see examples below.

Examples

```
#Tematres URLs
rtematres.options('tematres_url')
rtematres.options(tematres_url="http://www.example.com")
```

---

rtematres.search *Search for concepts*

Description

Search the thesaurus for concepts. This function is a wrapper and so it calls the appropriate funtions depending on the search task.

Usage

```
rtematres.search(term, includenotes = FALSE)
```

Arguments

term	The concept you are looking for
includenotes	Include definition texts in the search (true, false). Note that one is only working for searches containing > 2 characters. otherwise it is just ignored.

Value

The function returns a vecor or a list of results for the search

---

rtematres.summary *Create a summary for a vector*

Description

Search the thesaurus for concepts in case of categorical textual vector. Create a five value summary for numerical columns.

Usage

```
rtematres.summary(input)
```

Arguments

input	A categorial vector

Value

The function returns a summary

Examples

```
## Not run:
    rtematres.summary(input = iris$Species)
    lapply(iris, function(x) rtematres.summary(input = x))

## End(Not run)
```

Index