Package 'rols'

April 23, 2016

Type Package

Title An R interface to the Ontology Lookup Service

Version 1.12.2

Author@R person(given = ``Laurent'', family = ``Gatto'', email = ``lg390@cam.ac.uk'', role = c(``aut'',``cre''))

Author Laurent Gatto <lg390@cam.ac.uk>

Maintainer Laurent Gatto <lg390@cam.ac.uk>

Description This package allows to query EBI's Ontology Lookup Service (OLS) using Simple Object Access Protocol (SOAP).

Depends methods

Imports XML, XMLSchema (>= 0.6.0), SSOAP (>= 0.8.0), Biobase, utils

Suggests xtable, GO.db, knitr (>= 1.1.0), BiocStyle, testthat

biocViews Software, Annotation, MassSpectrometry, GO

License GPL-2

URL http://lgatto.github.com/rols/

Collate AllClasses.R AllGenerics.R iface.R cvparam.R environment.R queries.R rols-package.R utils.R zzz.R

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`rols-package` *R OLS interface*

Description

An R interface to the Ontology Lookup Service

Details

The Ontology Lookup Service (OLS) is a spin-off of the PRIDE project. It provides a web service interface to query multiple ontologies from a single location with a unified output format. This package allows to query the OLS from within R.

Author(s)

Laurent Gatto <lg390@cam.ac.uk>

References

http://www.ebi.ac.uk/ontology-lookup/

allIds *Returns all identifiers and terms of an ontology*

Description

This function returns all identifiers and terms available for given valid ontology. It sends a `getAllTermsFromOntologyReques` SOAP message and retrieves and parses the `getAllTermsFromOntologyResponse`. The original interface is `public Map getAllTermsFromOntology(String ontologyName)`.

Usage

```
allIds(ontologyName, simplify = TRUE)
```

Arguments

`ontologyName`	A `character` with the name of a valid ontology name.
`simplify`	A logical indicating whether the S4 `Map` instance should be simplified. Default is `TRUE`.

Value

A named `character` if `simplify` is `TRUE`. An instance of class `Map` otherwise.

Author(s)

Laurent Gatto

See Also

Other ols-queries: `isIdObsolete`; `olsQuery`; `olsVersion`; `ontologies`; `ontologyLoadDate`; `ontologyNames`; `rootId`; `termMetadata`; `termXrefs`; `term`

Examples

```
allIds("MS", simplify=FALSE)
```

as.character.Map *Coerce* Map *to a* data.frame

Description

as method to coerce an instance of Map to a character. The maps keys are used to name the map values.

Usage

```
## S3 method for class 'Map'
as.character(x, ...)
```

Arguments

x	An instance Map.
...	not used.

Value

A character of length length(map).

Author(s)

Laurent Gatto

See Also

Other as: as.character.mapItem

as.character.mapItem *Coerce* mapItem *to a* character

Description

as method to coerce an instance of mapItem to a character by concatenating the key and value variabels.

Usage

```
## S3 method for class 'mapItem'
as.character(x, ...)
```

Arguments

x	An instance of mapItem.
...	not used

Value

A character of length 2.

Author(s)

Laurent Gatto

See Also

Other as: as.character.Map

---

childrenRelations *Returns the children relation type(s).*

Description

This function returns the relation type of a ontology term termId and its children. The function sends a getTermRelationsRequest SOAP message and retrieves and parses the getTermRelationsResponse. The original corresponging interface is public Map getTermRelations(String termId, String ontologyName).

Usage

```
childrenRelations(termId = termId, ontologyName = ontologyName,
  simplify = TRUE)
```

Arguments

termId	A character with a valid ontology identifier.
ontologyName	A character with the name of a valid ontology name.
simplify	A logical indicating whether the S4 Map instance should be simplified. Default is TRUE.

Value

A named character if simplify is TRUE. An instance of class Map otherwise.

Author(s)

Laurent Gatto

Examples

```
childrenRelations("GO:0005802", "GO")
```

CVParam-class *Class* "CVParam"

Description

CVParam objects instantiate controlled vocabulary entries.

Usage

```
CVParam(label, name, accession, value, exact = TRUE)
```

Arguments

label	A character with the ontology label. If missing, a user-defined parameter is created.
name	A character with the name of the CVParam to be constructed. This argument can be omitted if accession is used and label is not missing.
accession	A character with the accession of the CVParam to be constructed. This argument can be omitted if name is used. Ignored for user-defined instances.
value	A character with the value of the CVParam to be constructed. This argument is optional.
exact	A logical defining whether the query to retrieve the accession (when name is used) should be an exact match.

Objects from the Class

Objects can be created with the CVParam constructor.

Slots

label: Object of class "character" that defines the label of the instance, i.e the ontology abbreviation. See ontologyNames for a complete list.

accession: Object of class "character" with the parameter's valid label ontology accession number. See below for validity constrains.

name: Object of class "character" with the instance's valid name, i.e matching with the accession. name and accession must follow term(accession, label) == name for the instance to be valid.

value: Object of class "character" with the CVParams value, if applicable, of empty string ("") otherwise.

user: Object of class "logical" defining if the instance is a user-defined parameter (also called User params).

.__classVersion__: Object of class "Versions" describing the instance's class definition version. For development use.

Extends

Class "Versioned", directly.

Methods

charIsCVParam(x) Checks if x, a character of the form "[ONTO, ACCESSION, NAME, VALUE]", is a valid (possibly user-defined) CVParam. "ONTO" is the ontology label (it must be one of ontologies()), "ACCESSION" is the term accession number, "NAME" is the term's name and "VALUE" is the value. Note that only synatex validity is verified, not semantics. See example below.

Methods

coerce signature(from = "CVParam", to = "character"): Coerces CVParam from to a character of the following form: [label, accession, name,value]. as.character is also defined.

coerce signature(from = "character", to = "CVParam"): Coerces character from to a CVParam. as.CVParam is also defined. If a label is absent, the character is converted to a User param, else, the label and accession are used to query the Ontology Lookup Service (see olsQuery)). If a name is provided and does not match the retrieved name, a warning is thrown.

This function is vectorised; if the from character is of length greater than 1, then a list of CVParam is returned. The queries to the OLS are processed one-by-one, though.

show signature(object = "CVParam"): Prints the CVParam instance as text.

rep signature(x = "CVParam", times = "numeric"): Replicates the CVParam x times times.

Author(s)

Laurent Gatto <lg390@cam.ac.uk>

Examples

```
## a user param
CVParam(name = "A user param", value = "the value")
## a CVParam from PSI's Mass Spectrometry ontology
term("MS:1000073", "MS")
CVParam(label = "MS", accession = "MS:1000073")
CVParam(label = "MS", name ="electrospray ionization")
CVParam(label = "MS", name ="ESI") ## using a synonym


## From a CVParam object to a character
cv <- as(CVParam(label = "MS", accession = "MS:1000073"), "character")
cv

## From a character object to a CVParam
as(cv, "CVParam")
as("[MS, MS:1000073, , ]", "CVParam") ## no name
as("[MS, MS:1000073, ESI, ]", "CVParam") ## name does not match
```

```
as(c(cv, cv), "CVParam") ## more than 1 character

x <- c("[MS, MS:1000073, , ]", ## valid CV param
       "[, , Hello, world]",   ## valid User param
       "[this, one is, not, valid]", ## bnot valid
       "[ , , , ]") ## not valid

stopifnot(charIsCVParam(x) == c(TRUE, TRUE, FALSE, FALSE))

## A list of expected valid and non-valid entries
rols:::validCVchars
rols:::notvalidCVchars
```

`isIdObsolete` *Is the ontology id obsolete*

Description

When terms are found to be outside the scope of an ontology, are misleadingly named or defined or describe a concept that would be better represented in another way, the terms are marked obsolete rather than deleted. This function tests this by sending an `isObsoleteRequest` SOAP message and retrieves and parses the `isObsoleteResponse`. The original interface is `public boolean isObsolete(String termId, St`

Usage

```
isIdObsolete(termId, ontologyName)
```

Arguments

`termId`	A `character` with a valid ontology identifier.
`ontologyName`	A `character` with the name of a valid ontology name.

Value

A `logical` specifying if the term id is obsolete.

Author(s)

Laurent Gatto

See Also

Other ols-queries: `allIds`; `olsQuery`; `olsVersion`; `ontologies`; `ontologyLoadDate`; `ontologyNames`; `rootId`; `termMetadata`; `termXrefs`; `term`

Examples

```
## is obsolete
term("GO:0005563", "GO")
isIdObsolete(termId = "GO:0005563", ontologyName = "GO")
stopifnot(isIdObsolete(termId = "GO:0005563", ontologyName = "GO"))
## replaced by
term("GO:0030533", "GO")
isIdObsolete(termId = "GO:0030533", ontologyName = "GO")
```

---

key,mapItem-method key *slot accessor.*

Description

key slot accessor for the mapItem instances.

key slot accessor for the Map instances.

Usage

```
## S4 method for signature 'mapItem'
key(object)

## S4 method for signature 'Map'
key(object)
```

Arguments

object	An instance of class Map or mapItem.

Value

A character.

A character.

Author(s)

Laurent Gatto

map,ANY-method — Map *slot accessor.*

Description

Accessor for the Map data of the OLS return messages converted to their respective S4 classes. The actual data is stored in Map slots.

Usage

```
## S4 method for signature 'ANY'
map(object)
```

Arguments

object	An S4 class produced by an OLS return message.

Value

A instance of class Map.

Author(s)

Laurent Gatto

olsQuery — *Returns matching identifiers*

Description

This function queries one or all ontologies for a pattern and returns all identifiers/terms. If a valid ontologyName is provided, only that ontology is queried. The function then sends a getTermsByNameRequest SOAP message and retrieves and parses the getTermsByNameResponse. The original corresponging interface is public Map getTermsByName(String partialName, String ontologyName, boolean reverseKeyOrde If no ontologyName is provided, all ontologies are used; the function then sends a getPrefixedTermsByNameRequest SOAP message and retrieves and parses the getPrefixedTermsByNameResponse. The original corresponging interface is public Map getPrefixedTermsByName(String partialName, boolean reverseKeyOrder).

Usage

```
olsQuery(pattern, ontologyName, exact = FALSE, n = 3, simplify = TRUE)
```

Arguments

pattern	A character used to query the OLS.
ontologyName	Optional. A character with the name of a valid ontology name. If missing, all ontologies are searched for pattern.
exact	Require pattern to match term exactly. Default is FALSE. Note that if ontologyName is missing, exact is ignored.
n	Number of attempts to repeat the query if no result is found. Default is 3.
simplify	A logical indicating whether the S4 Map instance should be simplified. Default is TRUE.

Details

Some valid queries sometimes return empty results due to network instabilities. For this reason, each olsQuery is repeated 3 times (see n parameter) as long as empty resuls are obtained. In general, when the ontology is specified, queries are fast and reliable.

Value

A named character if simplify is TRUE. An instance of class Map otherwise.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsVersion; ontologies; ontologyLoadDate; ontologyNames; rootId; termMetadata; termXrefs; term

Examples

```
olsQuery("tgn","GO") ## search GO for 'tgn'
olsQuery("tgn") ## search all ontologies
olsQuery("ESI", "MS")
olsQuery("ESI", "MS", exact = TRUE)
```

---

olsVersion *Returns the OLS version*

Description

This function returns the Ontology Lookup Webservice version, build data and author. It sends a getVersionRequest SOAP message and retrieves and parses the getVersionResponse. The original interface is public String getVersion().

Usage

```
olsVersion()
```

Value

A character of length 5.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsQuery; ontologies; ontologyLoadDate; ontologyNames; rootId; termMetadata; termXrefs; term

Examples

```
olsVersion()
```

---

ontologies *Returns all available ontologies*

Description

This function returns available ontologies. It sends a getOntologyNamesRequest SOAP message and retrieves and parses the getOntologyNamesResponse. The original interface is public Map getOntologyNames().

Usage

```
ontologies(simplify = TRUE)
```

Arguments

simplify	A logical indicating whether the S4 Map instance should be simplified. Default is TRUE.

Value

If simplify is TRUE, a data.frame with available ontologies names and descriptions. An instance of class Map otherwise.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsQuery; olsVersion; ontologyLoadDate; ontologyNames; rootId; termMetadata; termXrefs; term

Examples

```
head(ontologies())
ontologies(simplify=FALSE)
```

`ontologyLoadDate` *Returns the ontology load date*

Description

This function returns the load date of a given ontology. The ontology name must be valid, i.e. exists in `ontologies()`. It sends a `getOntologyLoadDateRequest` SOAP message and retrieves and parses the `getOntologyLoadDateResponse`. The original interface is `public String getOntologyLoadDate(String onto`

Usage

```
ontologyLoadDate(ontologyName)
```

Arguments

`ontologyName`	A `character` with the name of a valid ontology name.

Value

A `character` with the ontology's load date.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsQuery; olsVersion; ontologies; ontologyNames; rootId; termMetadata; termXrefs; term

Examples

```
ontologyLoadDate("GO")
ontologyLoadDate("FIX")
```

ontologyNames *Returns all ontologyNames*

Description

Returns the names of the OLS ontologies.

Usage

```
ontologyNames()
```

Value

A charcter with all ontology names.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsQuery; olsVersion; ontologies; ontologyLoadDate; rootId; termMetadata; termXrefs; term

Examples

```
head(ontologyNames())
```

parents *Returns the parent(s) of a term.*

Description

This function returns the parent term(s) of term termId in ontology ontologyName. The function sends a getTermParentsRequest SOAP message and retrieves and parses the getTermParentsResponse. The original corresponging interface is public Map getTermParents(String termId, String ontologyName)

Usage

```
parents(termId = termId, ontologyName = ontologyName, simplify = TRUE)
```

Arguments

termId	A character with a valid ontology identifier.
ontologyName	A character with the name of a valid ontology name.
simplify	A logical indicating whether the S4 Map instance should be simplified. Default is TRUE.

Value

A named `character` if `simplify` is `TRUE`. An instance of class `Map` otherwise.

Author(s)

Laurent Gatto

Examples

```
parents("GO:0005802", "GO")
```

---

rootId *Retuns the root identifiers of an ontology*

Description

This function returns root identifier(s) for a given valid ontology name. It sends a `getRootTermsRequest` SOAP message and retrieves and parses the `getRootTermsResponse`. The original interface is `public Map getRootTerms(String ontologyName)`.

Usage

```
rootId(ontologyName, simplify = TRUE)
```

Arguments

`ontologyName`	A `character` with the name of a valid ontology name.
`simplify`	A logical indicating whether the S4 `Map` instance should be simplified. Default is `TRUE`.

Value

A named `character` if `simplify` is `TRUE`. An instance of class `Map` otherwise.

Author(s)

Laurent Gatto

See Also

Other ols-queries: `allIds`; `isIdObsolete`; `olsQuery`; `olsVersion`; `ontologies`; `ontologyLoadDate`; `ontologyNames`; `termMetadata`; `termXrefs`; `term`

Examples

```
rootId("GO")
rootId("NEWT")
rootId("MS")
```

show,Map-method Map *show method*

Description

show method for Map instances

Usage

```
## S4 method for signature 'Map'
show(object)
```

Arguments

object	An Map instance.

Value

Returns an invisible 'NULL'. This function is used for its side-effect of printing a textual description of object.

Author(s)

Laurent Gatto

term *Returns the term of a given identifier*

Description

This function returns the term (description) of a given ontology identifier in a specific ontology. The ontology name must be valid, i.e. exists in ontologies(). It sends a getTermByIdRequest SOAP message and retrieves and parses the getTermByIdResponse. The original interface is public String getTermById(String termId, String ontologyName).

Usage

```
term(termId, ontologyName)
```

Arguments

termId	A character with a valid ontology identifier.
ontologyName	A character with the name of a valid ontology name.

Value

A string with the description of that identifier, as found in ontolgy ontologies. If termId was not found in ontologies(), as warning is issued and NULL is returned.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsQuery; olsVersion; ontologies; ontologyLoadDate; ontologyNames; rootId; termMetadata; termXrefs

Examples

```
term("GO:0005794", "GO") ## valid description
term("GO:0000000", "GO") ## returns NULL
term("210797", "NEWT")
```

---

termMetadata *Retuns an identifier's metadata*

Description

This function returns the metadata (definition and synonyms) for a specific ontology identifier. The term for that identifier can be retrieved with term. The function sends a getTermMetadataRequest SOAP message and retrieves and parses the getTermMetadataResponse. The original interface is public Map getTermMetadata(String termId, String ontologyName).

Usage

```
termMetadata(termId, ontologyName, simplify = TRUE)
```

Arguments

termId	A character with a valid ontology identifier.
ontologyName	A character with the name of a valid ontology name.
simplify	A logical indicating whether the S4 Map instance should be simplified. Default is TRUE.

Value

Am S3 instance of class termMetadata (for pretty printing) and character if simplify is TRUE. An instance of class Map otherwise.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsQuery; olsVersion; ontologies; ontologyLoadDate; ontologyNames; rootId; termXrefs; term

Examples

```
termMetadata("GO:0005794", "GO")
termMetadata("210797", "NEWT")
```

`termXrefs` *Returns the idenifier's ontology cross references*

Description

This function returns ontology cross references for an identifier. The function sends a `getTermXrefsRequest` SOAP message and retrieves and parses the `getTermXrefsResponse`. The original interface is `public Map getTermXrefs(String termId, String ontologyName)`.

Usage

```
termXrefs(termId, ontologyName, simplify = TRUE)
```

Arguments

`termId`	A `character` with a valid ontology identifier.
`ontologyName`	A `character` with the name of a valid ontology name.
`simplify`	A logical indicating whether the S4 `Map` instance should be simplified. Default is `TRUE`.

Value

A named `character` if `simplify` is `TRUE`. An instance of class `Map` otherwise.

Author(s)

Laurent Gatto

See Also

Other ols-queries: allIds; isIdObsolete; olsQuery; olsVersion; ontologies; ontologyLoadDate; ontologyNames; rootId; termMetadata; term

value,mapItem-method *value slot accessor.*

Description

value slot accessor for the mapItem instances.

value slot accessor for the Map instances.

Usage

```
## S4 method for signature 'mapItem'
value(object)

## S4 method for signature 'Map'
value(object)
```

Arguments

object	An instance of class Map or mapItem.

Value

A character.

A character.

Author(s)

Laurent Gatto

Index