

Package ‘datamart’

July 2, 2014

Type Package

Title Unified access to your data sources

Version 0.4.2

Date 2013-07-22

Author Karsten Weinert

Maintainer Karsten Weinert <k.weinert@gmx.net>

Description Provides an S4 infrastructure for unified handling of internal datasets and web based data sources. Examples include dbpedia, eurostat and sourceforge.

License GPL (>= 3)

Imports RJSONIO, XML, RCurl, base64, methods, markdown

LazyLoad yes

Collate '000_datamart-doc.r' '002_meta.R' '004_query.R'
'005_queries.R' '006_Xdata.R' '007_Res.R' '010_Twtr.R'
'020_Xsparql.R' '030_Dbpedia.R' '040_InternalData.R'
'045_Evs.R' '065_UrlData3.R' '080_iwv.R' '090_Mashup.R'
'110_SourceForge.R' '200_Eurostat.R' '212_UnitSetManager.R'
'230_Location.R' '231_DirectoryLocation.R' '232_Target.R'
'236_put.R' '237_FileTarget.R' '245_BlogPostTarget.R'
'250_Blogger.R' '252_MdFigure.R' '255_MdReport.R'
'260_SweaveReport.R' '270_CityCoords.R' '290_Pastebin.R'
'300_NetConnectGermany.R' '310_OddsPortal.r' 'localappdir.R' 'mem.info.R' 'strexts.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2013-07-23 02:12:14

R topics documented:

as.character	3
Blogger-class	3
BlogPostTarget-class	4
city_coords	4
datamart	5
Dbpedia-class	6
DirectoryLocation-class	6
Eurostat-class	7
Evs-class	8
FileTarget-class	9
InternalData-class	9
iwv_online	10
localappdir	10
Location-class	11
Mashup-class	11
MdFigure-class	12
MdReport-class	12
mem.info	13
MemoryLocation-class	14
meta	14
mySPARQL	15
netConnectGermany	15
OddsPortal-class	16
Pastebin-class	17
put	17
queries	18
query	18
Resource-class	19
show	19
siunit	20
sourceforge	21
strcap	21
strdecrypt	22
strencrypt	22
strhead	23
strparse	23
strrecode	24
strsubst	24
strtail	25
SweaveReport-class	25
Target-class	26
Twtrr-class	26
uconv	27
uconvlist	28
UrlData3-class	28
Xdata-class	29

<i>as.character</i>	3
Xsparql-class	29
Index	30

<i>as.character</i>	<i>The as.character for Xdata/Resource</i>
---------------------	--

Description

The *as.character* method has been defined for various classes, such as the *Resource* class and the *DirectoryLocation* class.

<i>Blogger-class</i>	<i>Location Class for Google's Blogger service</i>
----------------------	--

Description

This class implements a small subset of the blogger API. The authentication information (ClientLogin, no OAuth yet) and the title of the blog are passed to the constructor *blogger*. The meta method provides information on the submitted blogposts. The *put* method accepts a *BlogPostTarget* that can be transferred to *Blogger*.

Instantiates an object and authenticates with google.

Usage

```
blogger(blogtitle = getOption("blogger.blog"),
  email = getOption("blogger.username"),
  password = getOption("blogger.password"),
  class = "Blogger")
```

Arguments

<i>blogtitle</i>	name of the (existing) blog. Defaults to <code>getOption("blogger.blog")</code> .
<i>email</i>	email adress for authentication. Defaults to <code>getOption("blogger.username")</code> .
<i>password</i>	password for authentication. Defaults to <code>getOption("blogger.password")</code> .
<i>class</i>	name of the class for convenient inheritance. Defaults to "Blogger".

Value

Blogger

Examples

```
getSlots("Blogger")
```

BlogPostTarget-class *A Target representing a blog post*

Description

This is an internal class representing a blog post. Use MdReport instead.
For internal use only

Usage

```
blogpost(name, content, label = "", draft = TRUE,
         overwrite = TRUE, class = "BlogPostTarget")
```

Arguments

name	title of the blogpost
content	content of the post
label	character vector of keywords
draft	draft or not? default=TRUE
overwrite	overwrite or not? overwrite=TRUE
class	class of the object, default 'BlogPostTarget'

Value

BlogPostTarget

Examples

```
getSlots("BlogPostTarget")
```

city_coords *Longitude and Latitude for Cities*

Description

Based on Stackoverflow solution by Jochem Donkers.

Usage

```
city_coords(country = "DE")
```

Arguments

country	two-character country code, default 'DE'
---------	--

Value

an object of class `UrlData`

References

<http://stackoverflow.com/questions/13905098/how-to-get-the-longitude-and-latitude-coordinates-from>

datamart

Unified interface to your data sources.

Description

This package provides several S4 classes that make it easier to collect and work with datasets. The package is inspired by the [scraperwiki project](#), which provides a webbased service for data collection. Also inspiring are [Mathematica's xxxData functions](#), which provide in-built parametrizable datasets.

Details

You can specify web resources with the `urldata3` and the `xsparql` functions. You can combine several resources with the `mashup` function. For working with locally saved data, see the `internalData` function. The objects instantiated with these functions can then be passed to the generic query along with some parameters to get to the data.

Besides parameterized queries ("read" operations), the package also aims to support "write" operations. For this purpose, some functions (currently `mdreport`, `swvreport`) for defining targets as well as some functions (currently `blogger` and `dirloc`) for defining locations are provided. The generic `put` then builds the target and puts it at the defined location.

Some examples aim to proof the concept, for instance `eurostat_web`, `sourceforge`, `oddsportal`, and `city_coords`.

The package is highly experimental, and likely to change heavily without backward compatibility.

References

Karsten Weinert, [factbased blogspot](#).

Dbpedia-class *A class for querying Dbpedia.org*

Description

This class defines some resources at dbpedia. See `queries(dbpedia())` for a list of resources.

Constructor for Dbpedia objects

Usage

```
dbpedia(lang = "")
```

Arguments

lang two-character language code for the dbpedia, default ""

Value

a Dbpedia object, inherited from Xsparql

Examples

```
## Not run:
dbp <- dbpedia()
queries(dbpedia)
query(dbp, "Nuts1")

## End(Not run)
```

DirectoryLocation-class
Directory location

Description

The show method for the DirectoryLocation class has been adapted to display the path.

The dirloc function creates a DirectoryLocation object.

Usage

```
dirloc(path, class = "DirectoryLocation")
```

Arguments

path character, pointing to an existing directory. Required.
class character, optional class name. Default is "DirectoryLocation".

Details

The `as.character` method for the `DirectoryLocation` class returns the path to the directory it represents.

The `meta` method for the `DirectoryLocation` class returns the output of `file.info` of the folder.

Examples

```
getSlots("DirectoryLocation")
```

Eurostat-class

Eurostat web object.

Description

From Eurostat's webpage (see references): You can download individual datasets or the complete database by using the bulk download facility. On the bulk download you will find: all information updated twice a day, at 11:00 and 23:00, (at this time the service is usually not available, K.W.) the datasets in tsv (tab separated values) (this is what is used by this R code, K.W.), dft and sdmx format, which can be easily used to import the data in a tool of your choice, a manual containing all detailed information on the bulkdownload facility, the table of contents that includes the list of the datasets available, the "dictionaries" of all the coding systems used in the datasets.

`eurostat_data` – creates `UrlData` object for accessing Eurostat datasets.

`eurostat_dicts` – creates `UrlData` object for accessing Eurostat "dictionaries", i.e. a description of indicators.

`eurostat_toc` – creates `UrlData` object for accessing Eurostat "table of contents", i.e. a description of datasets.

Usage

```
eurostat_data()
```

```
eurostat_dicts()
```

```
eurostat_toc()
```

```
eurostat_web()
```

Details

The `eurostat_web()` function returns a data object with three methods, "EurostatToc" which returns a `data.frame` of available datasets, "EurostatDicts" which returns a large `data.frame` of codes, "EurostatData" which returns the dataset for a given code.

Value

an object of class `Mashup`

References

<http://epp.eurostat.ec.europa.eu/>

Examples

```
getSlots("Eurostat")
```

Evs-class

German Income and Expenditure Survey 2008 on private spendings, differentiated by household type and household income.

Description

Germany's Sample survey of income and expenditure (Einkommens- und Verbrauchsstichprobe, EVS) is conducted by the Federal Statistical Office. The data provided here is processed and is not identical with der Federal Office' data. Some information is lost by the processing. If you want more and/or more accurate data, contact the Federal Statistical Office.

Constructor for Evs data source class

Usage

```
expenditures()
```

Value

an object of class "Evs"

References

Statistisches Bundesamt: Wirtschaftsrechnungen. Einkommens- und Verbrauchsstichprobe. Einnahmen und Ausgaben privater Haushalte. Fachserie 15 Heft 4.

Examples

```
xp <- expenditures()
queries(xp)
query(xp, "Categories")
query(xp, "Elasticity", categ="05")
```

FileTarget-class	<i>FileTarget</i>
------------------	-------------------

Description

This class is a decorator for a physical existing file. A common workflow for other targets is to create a temporary file and then call put again with a FileTarget.

see class FileTarget for details.

Usage

```
filetarget(name, filename, class = "FileTarget")
```

Arguments

name	name of the Report, default ""
filename	name of original file
class	class name, default 'FileTarget'

Value

generic

Examples

```
getSlots("FileTarget")
```

InternalData-class	<i>A class for querying data()sets</i>
--------------------	--

Description

This class allows to query datasets that can be loaded with data(). Only read-only access.

Constructor for InternalData objects

Usage

```
internalData(name, package = NULL, class = "InternalData")
```

Arguments

name	name of the dataset. Required.
package	name of the package where the dataset is located. Default NULL.
class	name of the class to create. Default InternalData, must be inherited from this class.

Examples

```
getSlots("InternalData")
```

iwv_online	<i>website traffic from IWW</i>
------------	---------------------------------

Description

website traffic as tracked by iwv online. Use yyyyymm for query() as resource.

Usage

```
iwv_online()
```

```
iwv_online
```

Value

UrlData3 object

References

http://en.wikipedia.org/wiki/Informationsgemeinschaft_zur_Feststellung_der_Verbreitung_von_Werbetraegern

localappdir	<i>Platform-independent local app folder</i>
-------------	--

Description

determines local app folder based on Sys.info()["sysname"].

Usage

```
localappdir(appname = "R", create = TRUE)
```

Arguments

appname	subdir in local app folder, default "R"
create	logical, default TRUE, create folder if non-existent

Value

character

Location-class	<i>S4 base class to represent output Locations</i>
----------------	--

Description

The Location class is an abstract class that represents a place where to put something.

Mashup-class	<i>Mashup – A class for combining Xdata objects</i>
--------------	---

Description

This class administers a list of Xdata objects and delegates query and other methods to them. The aim of the class is to support the operation of combining different data sources into one data object.

Constructor for Mashup objects

Usage

```
mashup(..., class = "Mashup")
```

Arguments

...	named arguments of Xdata objects
class	name of the class to create. Default Mashup, must be inherited from this class.

Details

Instances of this class also maintain a database connection for storing/collecting data. The database connection is shared among the administered Xdata instances.

Usually it is not necessary for inherited classes to redefine the queries method. Rather, it is recommended to add query methods with new signatures in the resource slot.

Examples

```
getSlots("Mashup")
```

MdFigure-class *MdFigure*

Description

Internal Class representing figures in MdReport
 Internal function to create an MdFigure object.

Usage

```
mdfigure(name, xdata, resource = name, class = "MdFigure")
```

Arguments

name	name of the figure, default ""
xdata	call for creating the figure
resource	name of the resource
class	class name, default 'MdFigure'

Value

MdFigure

Examples

```
getSlots("MdReport")
```

MdReport-class *Buildable report*

Description

The main S4 class in this framework is MdReport. You can create a report with `mdreport`, which takes a template file name, and a list of variables as arguments. The new generic method `put` can then be used to actually produce the report at various locations (directory, memory, blogging site). see class MdReport for details.

Usage

```
mdreport(tpl, name = "", xdata = NULL, class = "MdReport",
  verbose = getOption("verbose"), ...)
```

Arguments

tpl	path to markdown template file
name	name of the Report, default ""
xdata	instance of Xdata class
verbose	diagnostic messages T/F
class	class of the object, default 'MdReport'
...	number of targets

Details

strsubst is a simple templating mechanism inspired from Python (PEP-0292). Variables in the template are marked by a preceding dollar sign and get replaced with the value of the corresponding variables passed to strsubst.

Value

generic

Examples

```
getSlots("MdReport")
```

mem.info	<i>Improved list of objects</i>
----------	---------------------------------

Description

Improved list of objects

Usage

```
mem.info(env = .GlobalEnv, sortBy = "Size")
```

Arguments

env	the environment to inspect, default is .GlobalEnv
sortBy	the result will be decreasingly sorted by this column. Possible values "Type", "Size" (default), "Rows", "Columns"

Value

data.frame with object information

Author(s)

Petr Pikal, David Hinds and Dirk Eddelbuettel

References

<http://stackoverflow.com/questions/1358003/tricks-to-manage-the-available-memory-in-an-r-session>

MemoryLocation-class *S4 base class to represent output in Memory*

Description

The MemoryLocation class represents the place in the memory of the current R process.

Examples

```
getSlots("MemoryLocation")
```

meta *Verbose list of resources*

Description

The meta method returns a data.frame with meta information entities available at the location. By default, zero rows are returned.

Arguments

self an Location object

Details

Inherited classes should override this method if necessary.

mySPARQL	<i>Internal query method for SPARQL end points</i>
----------	--

Description

Internal function, use `query(xsparql(), ...)` instead.

Usage

```
mySPARQL(url, query, typeconv = TRUE,
          verbose = getOption("verbose"))
```

Arguments

<code>url</code>	URL of SPARQL end point
<code>query</code>	SPARQL statement
<code>typeconv</code>	if TRUE (default), converts numbers and dates to R types
<code>verbose</code>	if TRUE, print diagnostic messages. Defaults to <code>getOption("verbose")</code>

Value

data.frame object

Author(s)

see SPARQL package

netConnectGermany	<i>Basic Price Information on natural Gas in Germany</i>
-------------------	--

Description

Basic Price Information on natural Gas in Germany

Usage

```
netConnectGermany(class = "UrlData3")
```

Arguments

<code>class</code>	Class name for the object, default <code>UrlData2</code>
--------------------	--

Value

UrlData2

References

http://datenservice.net-connect-germany.de/Dokumente/NCG_XML_Interface_V1.6_de.pdf

OddsPortal-class	<i>Query bet quotes from oddsportal</i>
------------------	---

Description

This class implements a "HTML API" to the oddsportal. The `queries` function returns the available API commands (currently only the German premier soccer league).

Constructor for OddsPortal class

Usage

```
oddsportal(class = "OddsPortal")
```

Arguments

<code>class</code>	Class name for the object, default OddsPortal
--------------------	---

Details

The `query` function constructs the URL(s), downloads and parses the data, and returns a `data.frame` of the result.

Value

OddsPortal

Examples

```
getSlots("OddsPortal")
```


Pastebin-class *Pastebin*

Description

This class exposes partially the Web API to the pastebin service.
 see Pastebin class for more information

Usage

```
pastebin(api_dev_key = getOption("pastebin.api_dev_key"),
         api_user_name = getOption("pastebin.api_user_name"),
         api_user_password = getOption("pastebin.api_user_password"),
         cls = "Pastebin")
```

Arguments

api_dev_key API Dev Key, default getOption("pastebin.api_dev_key")
 api_user_name API User Name, default getOption("pastebin.api_user_name")
 api_user_password API User password, default getOption("pastebin.api_user_password")
 cls Class name to initiate, default "Pastebin"

Examples

```
getSlots("Pastebin")
```

put *Put a target*

Description

This generic creates the target at the given location.

Arguments

target an object of class Target or derived
 where an object of class Location or derived

queries	<i>List resources</i>
---------	-----------------------

Description

The `queries` method returns a character vector of all defined resources for the given data object.

Arguments

<code>self</code>	an Xdata object
-------------------	-----------------

Details

The default (XData) implementation inspects definitions of the `query` method. Inherited classes should override this method if necessary.

query	<i>Request data from data source</i>
-------	--------------------------------------

Description

This generic function is the main interface to the data behind the Xdata layer. The first argument is the data object, the second argument is an identifier, usually a string, of the resource requested.

Arguments

<code>self</code>	an Xdata object
<code>resource</code>	an identifier of the resource requested. End-user usually provide character, developer use <code>resource</code> and dispatch on the type.
<code>...</code>	additional parameter

Details

Depending on the data object, additional parameter can be provided.

The slot `resource` should be filled with the help of the `resource` function. The default (Xdata) implementation with `resource` as character creates a `Resource` object of that name and calls itself again with this parameter. Thus dispatching on the resource is possible.

Resource-class	<i>S4 base class to represent resources</i>
----------------	---

Description

This class is used as based class when defining new resources for the query method. The resource function takes a character and defines a S4 class derived from Resource. Thus dispatching on the signature is possible, which is the main reason for using this class.

The resource function takes a character as input. If no class with this name exists, the function calls setClass and creates one, derived from Resource. The function returns its input, i.e. its use lies in its side effect.

Usage

```
resource(id, verbose = getOption("verbose"))
```

Arguments

id	character, identifier of the resource
verbose	if TRUE, diagnostic messages. Defaults to getOption("verbose").

Details

Usually, you do not use the class directly. The exception is to use the resource function in the signature argument of setMethod when defining your own queries.

Examples

```
getSlots("Resource")
```

show	<i>Show Method for Xdata classes</i>
------	--------------------------------------

Description

The show method for the Xdata class has been adapted to display the class name. Some inherited classes such DirectoryLocation or Blogger override this default definition.

`siunit`*Generate numeric vector with SI prefixes*

Description

Given a basename for the unit, e.g. "m", the function returns a named vector of most or all (extended=TRUE) prefixes defined by SI, i.e. "km", "Mm", "Gm", ... and "mm", "nm", ...

Usage

```
siunit(uname, value = 1, extended = FALSE)
```

Arguments

<code>uname</code>	the name of the unit to define, e.g. "g"
<code>value</code>	scaling factor, see details.
<code>extended</code>	logical (default=FALSE), add not so common prefixes?

Details

By default, the prefixes "h", "d", "da" and "c" are not generated. Use extended=TRUE to include these.

Use `value` to define a conversion factor between the unit without prefix and the base unit of the unitset you are defining. `value` is how many units of `uname` form one unit of the base unit. Default is 1.

The function is internal.

Value

numeric

References

http://en.wikipedia.org/wiki/International_System_of_Units

`sourceforge`*SourceForge – query stats for sourceforge projects*

Description

SourceForge – query stats for sourceforge projects

Usage

```
sourceforge(proj, from = "2008-01-01", class = "UrlData3")
```

Arguments

<code>proj</code>	name of the project
<code>from</code>	when did the project start? Default "2008-01-01".
<code>class</code>	which class to instantiate, default "UrlData3"

References

<http://sourceforge.net/p/forge/documentation/>

`strcap`*Capitalize a string*

Description

The first character is uppercased, the other lowercased.

Usage

```
strcap(s)
```

Arguments

<code>s</code>	character or character vector
----------------	-------------------------------

Value

character

`strdecrypt`*Obfuscate string*

Description

reverts the action of strencrypt (not safe across machines!)

Usage

```
strdecrypt(message)
```

Arguments

message character or character vector

Value

character

`strencrypt`*Obfuscate string*

Description

a fancy method to make the string unreadable use strdecrypt to revert (not safe across machines!)

Usage

```
strencrypt(message)
```

Arguments

message character or character vector

Value

character

strhead	<i>Get the first n letters</i>
---------	--------------------------------

Description

if $n > 0$, return the first n letters of x if $n < 0$, return all but the last $\text{abs}(n)$ letters of x

Usage

```
strhead(s, n = 1)
```

Arguments

s	character or character vector
n	numeric, default 1

Value

character

strparse	<i>Parse named patterns</i>
----------	-----------------------------

Description

code based on examples for `regexpr()`

Usage

```
strparse(pat, x)
```

Arguments

x	character or character vector
pat	named pattern

Value

named character vector or matrix

strrecode *Pattern-based recoding*

Description

Pattern-based recoding

Usage

```
strrecode(pats, repls, x, ...)
```

Arguments

pats	vector of patterns
repls	vector of replacements
x	character or character vector
...	additional parameter, passed to grepl

Value

replaced vector

strsubst *Named substitution in strings*

Description

Simple template mechanism inspired by PEP-0292. Use lists or named character vectors (vectors not tested) as a mapping for substitution.

Usage

```
strsubst(template, map, verbose = getOption("verbose"))
```

Arguments

template	character with \$(VARS)
map	object with [functionality e.g. a vector. Should return values that can be coerced to character
verbose	print debugging messages when TRUE, default is getOption("verbose")

Details

Substitutions are marked by \$(NAME).

Value

character

References

<http://www.python.org/dev/peps/pep-0292/> <http://stackoverflow.com/questions/8703398/conditional-gsub-replacement/8703832#8703832>

 strtail

Get the last n letters

Description

if n>0, return the last n letters of x if n<0, return all but the first abs(n) letters of x

Usage

```
strtail(s, n = 1)
```

Arguments

s	character or character vector
n	numeric, default 1

Value

character

 SweaveReport-class

Wrapper for Sweave and pdf

Description

The main S4 class in this framework is SweaveReport. You can create a report with swvreport, which takes a sweave file name, The generic method put can then be used to actually produce the report in pdf format.

see class SweaveReport for details.

Usage

```
swvreport(tpl, name = NULL, class = "SweaveReport",
  verbose = getOption("verbose"), ...)
```

Arguments

tpl	path to markdown template file
name	name of the Report, default ""
verbose	diagnostic messages T/F
class	class of the constructed object, default 'SweaveReport'
...	number of targets

Value

generic

Author(s)

Karsten Weinert <k.weinert@gmx.net>

Examples

```
getSlots("SweaveReport")
```

Target-class

Buildable target

Description

Buildable target

Twtrr-class

A class for querying twitter

Description

This class allows to query user timelines. Other resources have not been implemented yet. The class is subject to change in further versions.

Use `twtrr()` to initialize. If no user parameter are provided, then the global options `pft.user` is used.

Usage

```
twtrr(user = "")
```

Arguments

user	character, twitter screenname
------	-------------------------------

Examples

```
## Not run:
tw <- twttr(user="karstengweinert")
query(tw, "User_timeline")

## End(Not run)
```

uconv

Convert between numerical units

Description

This function converts between numerical units. It works similar to the `iconv` function: You provide vector `x` and a `from` and a `to` unit name and the function converts.

This is an internal class. It administers the unitsets used by the `uconv` method. One instance, usually the only one, is created at startup.

Internal function to create an `UnitSetManager` object.

Usage

```
uconv(x, from, to, uset = NULL)

unitsetmanager()
```

Arguments

<code>x</code>	numerical vector
<code>from</code>	character, unit to convert from.
<code>to</code>	character, unit to convert to
<code>uset</code>	optional, character, unit set to use.

Details

Additionally, you may provide a unitset name. Here, the analogy to `iconv` ceases. Think of unitset as a dimension of units, or a context for units. Predefined unitsets are "Length", "Mass", "Energy", and "Temperature". It is recommended to provide the unitset name. A list of available unitsets and the units defined by them can be obtained with `uconvlist()`.

Examples

```
uconv(1, "horse length", "m", "Length")
uconv(1:10, "TWh", "PJ", "Energy")
getSlots("UnitSetManager")
```

uconvlist	<i>List unitsets and their units</i>
-----------	--------------------------------------

Description

The function lists the currently available unitsets and the units supported by them.

Usage

```
uconvlist()
```

Value

named list, names=Unitsets, values=Units in these Unitsets

UrlData3-class	<i>UrlData3 – unified access to WWW resources</i>
----------------	---

Description

This class provides the infrastructure to scrape the web with a Extract, Transform, Load (ETL) approach.

Constructor for UrlData3 objects

Usage

```
urldata3(resource, template, extract.fct = readLines,
         transform.fct = identity, class = "UrlData3", ...)
```

Arguments

resource	the name of the resource. Required.
template	a pattern for the url. Must contain for substitution. Required.
extract.fct	a function that takes an URI and returns the raw data. Default readLines.
transform.fct	a function that takes the raw data and returns the cleaned/transformed data. Default identity.
class	name of the class to create. Default UrlData, must be inherited from this class.
...	parameters for the query. Must be named arguments, values can be characters (for defaults), NULL, or functions.

Details

In most cases, it is not necessary to subclass UrlData. The slots can be set by the urldata function and allow to customize each step of the process.

Examples

```
getSlots("UrlData3")
```

Xdata-class

Xdata – A class representing a data source

Description

Most methods of the class are abstract, however the show, print, queries methods will usually not need to be redefined.

Details

The query method is defined for character resource arguments. It is tried to transform the argument to an object; if that succeeds, query is called again. Derived methods that are also interpreting resource as character should first call this method via `callNextMethod`.

Xsparql-class

A class for querying SPARQL end points

Description

This class allows to run SELECT statement on SPARQL endpoints. The resource parameter is interpreted as SPARQL statement.

The function `xsparql` constructs a Xsparql object.

Usage

```
xsparql(url, nspace = "")
```

Arguments

url	sparql end point
nspace	character vector with short name / namespace expansions

Details

See Dbpedia for examples.

Value

a xsparql object

Examples

```
getSlots("Xsparql")
```

Index

*Topic **datasets**

- iwv_online, 10
- as.character, 3
- as.character,DirectoryLocation-method (as.character), 3
- as.character,Resource-method (as.character), 3
- blogger (Blogger-class), 3
- Blogger-class, 3
- blogpost (BlogPostTarget-class), 4
- BlogPostTarget-class, 4
- city_coords, 4
- datamart, 5
- datamart-package (datamart), 5
- dbpedia (Dbpedia-class), 6
- Dbpedia-class, 6
- DirectoryLocation-class, 6
- dirloc (DirectoryLocation-class), 6
- Eurostat-class, 7
- eurostat_data (Eurostat-class), 7
- eurostat_dicts (Eurostat-class), 7
- eurostat_toc (Eurostat-class), 7
- eurostat_web (Eurostat-class), 7
- Evs-class, 8
- evs2008.lvl2 (Evs-class), 8
- expenditures (Evs-class), 8
- filetarget (FileTarget-class), 9
- FileTarget-class, 9
- internalData (InternalData-class), 9
- InternalData-class, 9
- iwv_online, 10
- localappdir, 10
- Location-class, 11
- mashup (Mashup-class), 11
- Mashup-class, 11
- mdfigure (MdFigure-class), 12
- MdFigure-class, 12
- mdreport (MdReport-class), 12
- MdReport-class, 12
- mem.info, 13
- MemoryLocation-class, 14
- meta, 14
- meta,Blogger-method (meta), 14
- meta,DirectoryLocation-method (meta), 14
- meta,Pastebin-method (meta), 14
- meta,Xdata-method (meta), 14
- mySPARQL, 15
- netConnectGermany, 15
- oddsportal (OddsPortal-class), 16
- OddsPortal-class, 16
- pastebin (Pastebin-class), 17
- Pastebin-class, 17
- put, 17
- put,BlogPostTarget,Blogger-method (put), 17
- put,FileTarget,DirectoryLocation-method (put), 17
- put,MdFigure,DirectoryLocation-method (put), 17
- put,MdFigure,MemoryLocation-method (put), 17
- put,MdReport,Location-method (put), 17
- put,SweaveReport,DirectoryLocation-method (put), 17
- put,SweaveReport,missing-method (put), 17
- put,Target,character-method (put), 17
- queries, 18
- queries,InternalData-method (queries), 18

- queries,Location-method (queries), 18
- queries,Mashup-method (queries), 18
- queries,UrlData3-method (queries), 18
- queries,Xdata-method (queries), 18
- query, 18
- query,Dbpedia,Nuts1-method (query), 18
- query,Dbpedia,PlzAgs-method (query), 18
- query,Evs,Categories-method (query), 18
- query,Evs,Elasticities-method (query), 18
- query,Evs,Elasticity-method (query), 18
- query,InternalData,character-method (query), 18
- query,Mashup,character-method (query), 18
- query,Pastebin,character-method (query), 18
- query,Twtr,User_timeline-method (query), 18
- query,UnitSetManager,Energy-method (query), 18
- query,UnitSetManager,Length-method (query), 18
- query,UnitSetManager,Mass-method (query), 18
- query,UnitSetManager,Temperature-method (query), 18
- query,UrlData3,character-method (query), 18
- query,Xdata,character-method (query), 18
- query,Xsparql,character-method (query), 18

- resource (Resource-class), 19
- Resource-class, 19

- show, 19
- show,Blogger-method (show), 19
- show,DirectoryLocation-method (show), 19
- show,FileTarget-method (show), 19
- show,Pastebin-method (show), 19
- show,Target-method (show), 19
- show,Xdata-method (show), 19
- siunit, 20
- sourceforge, 21
- strcap, 21
- strdecrypt, 22
- strencrypt, 22
- strhead, 23

- strparse, 23
- strrcode, 24
- strsubst, 24
- strtail, 25
- SweaveReport-class, 25
- swvreport (SweaveReport-class), 25

- Target-class, 26
- twtr (Twtr-class), 26
- Twtr-class, 26

- uconv, 27
- uconvlist, 28
- unitsetmanager (uconv), 27
- UnitSetManager-class (uconv), 27
- urldata3 (UrlData3-class), 28
- UrlData3-class, 28

- Xdata-class, 29
- xsparql (Xsparql-class), 29
- Xsparql-class, 29