Package ‘stacomirtools’

October 14, 2022

Version 0.6.0.1
Date 2022-04-28
Title Connection Class for Package stacomIR
Author Cedric Briand [aut, cre],
      Marion Legrand [aut],
      Beaulaton Laurent [ctb]
Maintainer Cedric Briand <cedric.briand00@gmail.com>
Description S4 class wrappers for the 'ODBC' and Pool DBI connection, also provides some
      utilities to paste small datasets to clipboard, rename columns. It is used by the package 'stacomIR' for
      connections to the database. Development versions of 'stacomIR' are available in R-forge.
License GPL (>= 2)
Collate 'ConnectionODBC.R' 'RequeteODBC.R' 'RequeteODBCwhere.R'
      'RequeteODBCwheredate.R' 'utilities.R' 'stacomIRtools.R'
      'stacomIRtools-package.R' 'ConnectionDB.R' 'RequeteDB.R'
      'RequeteDBwhere.R' 'RequeteDBwheredate.R' 'zzz.R'
LazyLoad yes
Depends RODBC, DBI, pool
Imports methods, xtable, utils, RPostgres
Suggests testthat, withr
NeedsCompilation no
RoxygenNote 7.1.2
Repository CRAN
Date/Publication 2022-04-29 11:40:02 UTC

R topics documented:

  chnames .............................................................. 2
  connect ............................................................. 3
  connect,ConnectionDB-method ...................................... 4
  connect,ConnectionODBC-method ................................... 5
This function replaces the variable names in a data.frame

Usage

```r
chnames(object, old_variable_name, new_variable_name)
```

Arguments

- `object`: a data frame
- `old_variable_name`: a character vector with old variables names
- `new_variable_name`: a character vector with new variables names
**connect**

**Value**

object

**Author(s)**

Cedric Briand <cedric.briand@eptb-vilaine.fr>

**Examples**

```r
df <- data.frame("var1" = c("blue","red"), "var2" = c("nice","ugly"))
colnames(df) # "var1" "var2"
df <- chnames(object = df, old_variable_name = c("var1","var2"),
"new_variable_name" = c("color","beauty"))
colnames(df) # "color" "beauty"
# the following will return an error, as the variable wrong_name is not in variable names
## Not run:
chnames(object = df, old_variable_name = c("wrong_name"),
"new_variable_name" = c("color"))
## End(Not run)
```

---

**connect**

*generic connect function for baseODBC*

**Description**

*generic connect function for baseODBC*

**Usage**

```r
connect(object, ...)
```

**Arguments**

- **object**: an object
- **...**: additional arguments passed on to the connect method
connect,ConnectionDB-method

connect method for ConnectionDB class

Description

connect method for ConnectionDB class

Usage

```r
## S4 method for signature 'ConnectionDB'
connect(object, base = NULL)
```

Arguments

- `object`: An object of class ConnectionDB
- `base`: a string with values for dbname, host, port, user, password, in this order.

Value

a connection with slot filled

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

```r
## Not run:
object <- new("ConnectionDB")
object@dbname <- c("bd_contmig_nat","test")
object@host <- "localhost"
object@port <- "5432"
object@user <- "myuser"
object@password <- "mypassword"
object@silent=FALSE
object <- connect(object)
pool::dbGetInfo(object@connection)
pool::poolClose(object@connection)

## End(Not run)
```
connect,ConnectionODBC-method

connect method for ConnectionODBC class

Description
connect method for ConnectionODBC class

Usage
## S4 method for signature 'ConnectionODBC'
connect(object)

Arguments
object an object of class ConnectionODBC

Value
a connection with slot filled

Author(s)
Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples
##this wont be run as the user need to manually configure the connection before using it
## Not run:
object=new("ConnectionODBC")
object@baseODBC=c("myODBCconnection","myusername","mypassword")
object@silent=FALSE
object<-connect(object)
odbcClose(object@connection)

## End(Not run)

connect,RequeteODBC-method

connect method loads a request to the database and returns either an error or a data.frame

Description
connect method loads a request to the database and returns either an error or a data.frame
Usage

## S4 method for signature 'RequeteODBC'
connect(object)

Arguments

object an object of class RequeteODBC

Value

An object of class RequeteODBC

Note

assign("showmerequest",1,envir=envir_stacomi) allows to print all queries passing on the class RequeteDB

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

showClass("RequeteODBC")
## Not run:
object=new("RequeteODBC")
object@open=TRUE
object@baseODBC=baseODBC
object@sql= "select * from t_lot_lot limit 100"
object<-connect(object)
odbcClose(object@connection)
odbcCloseAll()

object=new("RequeteODBC")
object@open=TRUE
## this will leave the connection open,
## by default it closes after the query is sent
## the following will work only if you have configured and ODBC link
object@baseODBC=c("myODBCconnection","myusername","mypassword")
object@sql= "select * from mytable limit 100"
object<-connect(object)
odbcClose(object@connection)
envir_stacomi=new.env()
## While testing if you want to see the output of sometimes
## complex queries generated by the program
assign("showmerequest",1,envir_stacomi)
## You can assign any values (here 1)
## just tests the existence of "showmerequest" in envir_stacomi
object=new("RequeteODBC")
object@baseODBC=c("myODBCconnection","myusername","mypassword")
object@sql= "select * from mytable limit 100"
object<-connect(object)
## S4 method for signature 'RequeteODBCwhere'
connect(object)

### Arguments

- **object**: an object of class RequeteODBCwhere

### Description

connect method loads a request to the database and returns either an error or a data.frame

### Usage

```r
## Not run:
object<-new("RequeteODBCwhere")
object@baseODBC<-baseODBC
object@sql<="select * from t_lot_lot"
object@where<="WHERE lot_tax_code='2038'"
object@and<="c("AND lot_std_code='CIV','AND lot_ope_identifiant<1000")"
object@order_by<="ORDER BY lot_identifiant"
object<-connect(object)
## End(Not run)
```

### Examples

```r
## Not run:
object<-new("RequeteODBCwhere")
object@baseODBC<-baseODBC
object@sql<="select * from t_lot_lot"
object@where<="WHERE lot_tax_code='2038'"
object@and<="c("AND lot_std_code='CIV','AND lot_ope_identifiant<1000")"
object@order_by<="ORDER BY lot_identifiant"
object<-connect(object)
## End(Not run)
```
connect,RequeteODBCwheredate-method

connect method loads a request to the database and returns either an error or a data.frame

Description

connect method loads a request to the database and returns either an error or a data.frame

Usage

```r
## S4 method for signature 'RequeteODBCwheredate'
connect(object)
```

Arguments

- `object`: an object of class RequeteODBCwheredate

Value

An object of class RequeteODBCwheredate

Note

method modified from v0.2.1240 to use the connect method of the mother class which in turn will use the method of the mother class

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

```r
## Not run:
object<-new("RequeteODBCwheredate")
object@baseODBC<-get("baseODBC",envir=envir_stacomi)
object@select<- "select * from t_operation_ope"
object@datedebut=strptime("1996-01-01 00:00:00",format="%Y-%m-%d %H:%M:%S")
object@datefin=strptime("2000-01-01 00:00:00",format="%Y-%m-%d %H:%M:%S")
object@colonnedebut="ope_date_debut"
object@colonnefin="ope_date_fin"
object@and<-c("AND ope_dic_identifiant=1","AND ope_dic_identifiant=2")
object@order_by<="ORDER BY ope_identifiant"
object@silent=FALSE
object<-connect(object)

## End(Not run)
```
ConnectionDB-class

Description

ConnectionDB class

Value

connection an S4 object of class connectionDB

Slots

dbname name of the database, length 1
host host default "localhost", length 1
port port of the database default "5432", length 1
user user of the database, length 1
password password for the database, length 1
silent A "logical" stating if the program returns output to the user
status Can be -1 or string
connection Could be both string or S3

Note

Mother class for connection, opens the connection but does not shut it

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

```r
# this wont be run as the user need to manually configure the connection before using it
# Not run:
object <- new("ConnectionDB")
object$dbname <- c("bd_contmig_nat","test")
object$host <- "localhost"
object$port <- "5432"
object$user <- "myuser"
object$password <- "mypassword"
object$silent <- FALSE
object <- connect(object)
pool::dbGetInfo(object$connection)
pool::poolClose(object$connection)
```

## End(Not run)
ConnectionODBC-class  

Description

ConnectionODBC class

Value

connectionODBC an S4 object of class connectionODBC

Slots

baseODBC  "vector" (of length 3, character)
silent  "logical"
etat  "ANY" # can be -1 or string
connection  "ANY" # could be both string or S3

Note

Mother class for connection, opens the connection but does not shut it

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

##this wont be run as the user need to manually configure the connection before using it
## Not run:
object=new("ConnectionODBC")
object@baseODBC=c("myODBCconnection","myusername","mypassword")
object@silent=FALSE
object<connect(object)
odbcClose(object@connection)

## End(Not run)
ex fonction to write to the clipboard

Description

ex fonction to write to the clipboard

Usage

ex(d = NULL)

Arguments

d a dataframe

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

funhtml function used to print the html tables of output (see xtable documentation)

Description

see xtable for further description, an xtable is created and printed to html format

Usage

funhtml(
  data,
  caption = NULL,
  top = TRUE,
  outfile = NULL,
  clipboard = FALSE,
  append = TRUE,
  digits = NULL,
  ...
)

...
**getquery**

**Arguments**

- **data**  
  a data frame
- **caption**  
  the caption
- **top**  
  a logical, if true the caption is placed on top
- **outfile**  
  the path to the file
- **clipboard**  
  if clipboard TRUE, a copy to the clipboard is made
- **append**  
  is the file appended to the previous one?
- **digits**  
  the number of digits
- **...**  
  additional parameters to be passed to the function

**Value**

an xtable

**Author(s)**

Cedric Briand <cedric.briand@eptb-vilaine.fr>

---

**getquery**  
*generic query function for*

**Description**

generic query function for

**Usage**

getquery(object, ...)

**Arguments**

- **object**  
  an object
- **...**  
  additional parameters passed to query
getquery, RequeteDB-method

Description

getquery retrieves the result of the query from the object

Usage

## S4 method for signature 'RequeteDB'
getquery(object)

Arguments

object an object of class RequeteDB

Value

A data frame

induk unique values of a vector

Description

returns the index of values appearing only once in a vector: match(unique(a), a), replicated values are not returned on their second occurrence

Usage

induk(a)

Arguments

a a vector

Value

the index unique values within a vector

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>
is.even

Examples

induk(c(1,1,2,2,2,3))

is.even

is.even function modified from package sma

Description

is.even function modified from package sma (which did not verify that the entry was indeed an integer)

Usage

is.even(x)

Arguments

x integer

Value

a logical

Author(s)

Adapted from Henrik Bengtsson

Examples

is.even(1)
is.even(2)
is.odd  

*Description*

id.odd function modified from package sma (which did not verify that the entry was indeed an integer)

*Usage*

```r
is.odd(x)
```

*Arguments*

- `x` integer

*Value*

a logical

*Author(s)*

Adapted from Henrik Bengtsson

*Examples*

```r
is.odd(1)
is.odd(2)
```

---

killfactor  

*Description*

very usefull function used to "kill" the factors, noticeably after loading with 'ODBC'

*Usage*

```r
killfactor(df)
```

*Arguments*

- `df` a data.frame
query, RequeteDB-method

Value

df

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Description

generic query function for

Usage

query(object, ...)

Arguments

object an object
...
additional parameters passed to query

query, RequeteDB-method

query method loads a quert to the data and returns either an error or a data.frame

Description

query method loads a quert to the data and returns either an error or a data.frame

Usage

## S4 method for signature 'RequeteDB'
query(object, ...)

Arguments

object an object of class RequeteDB
...
进一步的参数传递给查询方法，base将被传递给ConnectionDB来设置连接参数，它应该是一个vector与dbname host port user和password（按此顺序）
query, RequeteDB where-method

Value

An object of class RequeteDB

Note

assign("showmerequest", 1, envir = envir_stacomi) allows to print all queries passing on the class RequeteDB

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

showClass("RequeteDB")
## Not run:
# this connection require user and password and a working data
object = new("RequeteDB", dbname = "bd_contmig_nat",
host = "localhost", port = "5432", user = "postgres", password = "secret")
object@open = TRUE
## this will leave the connection open,
## by default it closes after the query is sent
## the following will work only if you have configured and ODBC link
object$sql = "select * from t_lot_lot limit 100"
object$<query(object)
envir_stacomi = new.env()
## While testing if you want to see the output of sometimes
## complex queries generated by the program
assign("showmerequest", 1, envir_stacomi)
## You can assign any values (here 1)
## just tests the existence of "showmerequest" in envir_stacomi
object$sql = "select * from mytable limit 100"
object$<connect(object)
## the connection is already closed, the query is printed

## End(Not run)

query, RequeteDB where-method

query method loads a request to the database and returns either an error or a data.frame

Description

query method loads a request to the database and returns either an error or a data.frame
Usage

## S4 method for signature 'RequeteDBwhere'
query(object, ...)

Arguments

object an object of class RequeteDBwhere

... further arguments passed to the query method, base will be passed to ConnectionDB to set the connection parameters,

Value

An object of class RequeteODBCwhere

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

## Not run:
object<-new("RequeteODBCwhere")
base=c("bd_contmig_nat","localhost","5432","user","password")
object$ql<- "select * from t_lot_lot"
object$where<="WHERE lot_tax_code='2038'"
object$and<-c("AND lot_std_code='CIV'","AND lot_ope_identifiant<1000")
object$order_by<="ORDER BY lot_identifiant"
object <- connect(object, base)

## End(Not run)
RequeteDB-class

Arguments

object an object of class RequeteDBwheredate

... further arguments passed to the query method, base will be passed to ConnectionDB to set the connection parameters,

Value

An object of class RequeteDBwheredate

Author(s)

Cedric Briand <cedric.briand@eptb-vilaine.fr>

Examples

```r
## Not run:
object<-new("RequeteDBwheredate")
base=c("bd_contmig_nat","localhost","5432","user", "password")
object@select<- "select * from t_operation_ope"
object@datedebut=strptime("1996-01-01 00:00:00",format="%Y-%m-%d %H:%M:%S")
object@datefin=strptime("2000-01-01 00:00:00",format="%Y-%m-%d %H:%M:%S")
object@colonnedebut="ope_date_debut"
object@colonnefin="ope_date_fin"
object@and<-c("AND ope_dic_identifiant=1","AND ope_dic_identifiant=2")
object@order_by<="ORDER BY ope_identifiant"
object@silent=FALSE
object<-connect(object, base)
## End(Not run)
```

RequeteDB-class RequeteDB class

Description

RequeteDB class

Slots

sql "character"
query "data.frame"
open logical is the connection left open after the request ?

Note

Inherits from ConnectionDB
**Author(s)**

Cedric Briand <cedric.briand@eptb-vilaine.fr>

**Examples**

```r
object=new("RequeteDB")
```

---

**RequeteDBwhere-class**

**RequeteDBwhere class**

**Description**

RequeteDBwhere class

**Slots**

```r
select "character"
where "character"
and "vector"
order_by "character"
```

**Note**

Inherits from RequeteDB the syntax is where="WHERE ..." and =vector("AND...","AND...") order_by="ORDER BY.."

**Author(s)**

Cedric Briand <cedric.briand@eptb-vilaine.fr>

**Examples**

```r
object=new("RequeteDBwhere")
```
**RequeteDBwhere-date-class**

**RequeteDBwhere class**

**Description**

RequeteDBwhere class

**Slots**

- datedebut "POSIXlt"
- datefin "POSIXlt"
- colonnedebut "character" # name of the column containing datedebut
- colonnefin "character" # name of the column containing datefin

**Note**

Inherits from RequeteDBwhere and uses its connect method with a new SetAs

**Examples**

```r
object=new("RequeteDBwhere")
```

---

**RequeteODBC-class**

**RequeteODBC class**

**Description**

RequeteODBC class

**Slots**

- sql "character"
- query "data.frame"
- open logical is the connection left open after the request?

**Note**

Inherits from ConnectionODBC

**Author(s)**

Cedric Briand <cedric.briand@eptb-vilaine.fr>

**Examples**

```r
object=new("RequeteODBC")
```
**Description**

RequeteODBCwhere class

**Slots**

```r
selct "character"
where "character"
and "vector"
order_by "character"
```

**Note**

Inherits from RequeteODBC the syntax is where="WHERE ..." and =vector("AND...","AND...") order_by="ORDER BY.."

**Author(s)**

Cedric Briand <cedric.briand@eptb-vilaine.fr>

**Examples**

```r
object=new("RequeteODBCwhere")
```

**Description**

RequeteODBCwhere class

**Slots**

```r
datedebut "POSIXlt"
datefin "POSIXlt"
colonnebutent "character" # name of the column containing datedebut
colonnefin "character" # name of the column containing datefin
```
**Note**

Inherits from RequeteODBCwhere and uses its connect method with a new SetAs

**Examples**

```r
object = new("RequeteODBCwhere")
```

**Description**

Function to transform a ftable into dataframe but just keeping the counts, works with ftable of dim 2

**Usage**

```r
tab2df(tab)
```

**Arguments**

- `tab` a flat table

**Author(s)**

Cedric Briand <cedric.briand@eptb-vilaine.fr>

**Examples**

```r
df <- data.frame("var1" = c("blue","red"), "var2" = c("nice","ugly"))
ftdf <- ftable(df)
tab2df(ftdf)
```

---

**validity_DB**

**validity function for ConnectionDB class**

**Description**

validity function for ConnectionDB class

**Usage**

```r
validity_DB(object)
```

**Arguments**

- `object` An object of class ConnectionDB
validity_ODBC

---

validity_ODBC  

*Validity method for ODBC class*

---

**Description**

Validity method for ODBC class

**Usage**

`validity_ODBC(object)`

**Arguments**

- `object`  
  an object of class ConnectionODBC
Index

chnames, 2
connect, 3
connect,ConnectionDB-method, 4
connect,ConnectionODBC-method, 5
connect,RequeteODBC-method, 5
connect,RequeteODBCwhere-method, 7
connect,RequeteODBCwheredate-method, 8
connect.ConnectionDB
   (connect,ConnectionDB-method), 4
connect.ConnectionODBC
   (connect,ConnectionODBC-method), 5
connect.RequeteODBC
   (connect,RequeteODBC-method), 5
connect.RequeteODBCwhere
   (connect,RequeteODBCwhere-method), 7
connect.RequeteODBCwheredate
   (connect,RequeteODBCwheredate-method), 8
ConnectionDB-class, 9
ConnectionODBC-class, 10
query,RequeteDB-wheredate
   (query,RequeteDBwheredate-method), 18
query,RequeteODBCwhere
   (query,RequeteDBwhere-method), 17
RequeteDB-class, 19
RequeteDBwhere-class, 20
RequeteDBwheredate-class, 21
RequeteODBC-class, 21
RequeteODBCwhere-class, 22
RequeteODBCwheredate-class, 22
tab2df, 23
validity_DB, 23
validity_ODBC, 24