Package 'stashR'

February 20, 2015

Version 0.3-5

Date 2012-03-21

Depends R (>= 2.14.0), methods, filehash

Imports tools, digest

LazyLoad yes

Collate remoteDB.R localDB.R init.R util.R valid.R zzz.R

Title A Set of Tools for Administering SHared Repositories

Author Sandy Eckel, Roger D. Peng

Maintainer Roger D. Peng <rpeng@jhsph.edu>

Description A Set of Tools for Administering SHared Repositories

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-03-21 15:23:32

NeedsCompilation no

R topics documented:

---

stashR *Classes "remoteDB" and "localDB"*

Description

These functions create an interface for a simple file-based key-value database using remote storage via http

Objects from the "remoteDB" and "localDB" classes

Objects can be created by calls of the form new("remoteDB", ...) and new("localDB", ...) respectively.

Slots

url: Object of class "character", url of the remote database

dir: Object of class "character", local directory in which to download the data or local directory in which to create a database

Methods

dbDelete signature(db = "remoteDB", key = "character"): Calling dbDelete on a remoteDB object returns an error.

dbDelete signature(db = "localDB", key = "character"): Calling dbDelete on a localDB deletes the specified key from the database and updates the repository version.

dbExists signature(db = "remoteDB", key = "character"): For each element in the vector key, dbExists returns TRUE if the key is in the database and otherwise returns FALSE.

dbExists signature(db = "localDB", key = "character"): For each element in the vector key, dbExists returns TRUE if the key is in the database and otherwise returns FALSE.

dbFetch signature(db = "remoteDB", key = "character"): Checks if the provided character value key exists in the local directory. If not, dbFetch downloads the data files for the current version. Otherwise, dbFetch reads the file from the local directory. The function returns the data object associated with the key.

dbFetch signature(db = "localDB", key = "character"): Checks if the provided character value key exists in the local directory. If the key exists in the local directory, then dbFetch reads the file from the local directory. The function returns the data object associated with the key.

dbInsert signature(db = "remoteDB", key = "character", value = "ANY"): Calling dbInsert on a remoteDB object returns an error.

dbInsert signature(db = "localDB", key = "character",value = "ANY"): Calling dbInsert on a localDB object writes the value to a file corresponding to the specified key in the local directory and updates the version of the repository.

dbList signature(db = "remoteDB"): The method dbList returns a character vector of all the keys in the database.

dbList signature(db = "localDB"): The method dbList returns a character vector of all the keys in the database.

dbSync signature(db = "remoteDB", key = "character"): If key = NULL, Updates all key/data pairs in the local directory to the most recent repository version on the remote server. If key is a character vector, then it only updates the specified key/data pairs (in which case, it first checks to ensure that all specified keys' files have been previously saved).

dbCreate signature(db = "remoteDB"): Calling dbCreate on a remoteDB object creates the local main directory and data sub-directory for storing the data files and saves the url associated with the remoteDB object in the R workspace format in the local main directory. dbCreate is called implicitly when new is called to create the remoteDB object so calling dbCreate explicitly is usually not necessary.

dbCreate signature(db = "localDB"): Calling dbCreate on a localDB object creates the local main directory and data sub-directory to in which to store the data files. dbCreate is called implicitly when new is called to create the localDB object so calling dbCreate explicitly is usually not necessary.

Author(s)

Sandy Eckel, Roger D. Peng

Examples

```
## Not run:
## Objects of the class localDB

wd <- getwd()
dir <- file.path(wd,"localDBExample")

## Create local stashR data repository 'localDBExample'
fhLocal <- new("localDB", dir = dir, name = "localDBExample")

## Insert key-value data into 'localDBExample'
v <- 1:10
dbInsert(fhLocal,key = "vector", value = v)
m <- matrix(1:20,5,4)
dbInsert(fhLocal,key = "matrix", value = m)
d <- data.frame(cbind(id = 1:5, age=c(12,11,15,11,14), sex = c(1,1,0,1,0)))
dbInsert(fhLocal,key = "dataframe", value = d)
l <- list(v = v, m = m, df = d)
dbInsert(fhLocal, key = "list", value = l)

dbList(fhLocal)

dbFetch(fhLocal, "dataframe")
dbDelete(fhLocal, "vector")
dbExists(fhLocal, "vector")
dbList(fhLocal)


## Objects of the class remoteDB

## The same key-value data used in the previous example for localDB
## has been stored in a remoteDB repository on the internet at:
myurl <- "http://www.biostat.jhsph.edu/~seckel/remoteDBExample"

wd <- getwd()
dir <- file.path(wd,"remoteDBExample")
## Create local copy of data repository 'remoteDBExample'
fhRemote <- new("remoteDB", url= myurl,
            dir = dir, name= "remoteDBExample")

dbList(fhRemote)
dbExists(fhRemote,c("vector", "array","list", "function"))
```

```
## downloads 'vector' data from the remoteDB repository
dbFetch(fhRemote, "vector")
## fetches without downloading again
dbFetch(fhRemote, "vector")
## synchronize all local copies of the data to the remote version
dbSync(fhRemote, key = NULL)

## End(Not run)
```

---

stashROption   *Set options*

Description

Set options relevant to the stashR package

Usage

```
stashROption(name, value)
```

Arguments

name   character, name of option

value   value for the option

Details

Currently, the only option that can be set is quietDownload which is a logical indicating whether download progress should be printed to the screen when downloading a value associated with a specific key. The default is TRUE.

Author(s)

Roger D. Peng

---

utils   *Set directory for remote database*

Description

Utilities for "localDB" and "remoteDB" databases

Usage

```
setDir(db) <- value
copyDB(db, ...)
reposVersion(db, ...)
reposVersion(db) <- value

## S4 method for signature 'localDB'
initialize(.Object, ...)
## S4 method for signature 'remoteDB'
initialize(.Object, ...)

## S4 method for signature 'localDB'
copyDB(db, dir, ...)
```

Arguments

db	an object of class "remoteDB" or "localDB"
value	character, directory to be assigned
.Object	an object of class "localDB" or "remoteDB"
dir	directory into which the "localDB" should be copied
...	other arguments passed to methods

Details

Users obtaining "remoteDB" objects created on different machines may want to reset the directory specified for the local cache. The setDir replacement function is to be used for that purpose.

Users generally will not call the initialize functions directly, but rather will call them indirectly via the new function.

copyDB is used to copy a "localDB" database to another location (i.e. directory).

reposVersion returns the current repository version number and the replacement method can be used to set the repository to a different version number. A repository version of -1 indicates the most recent version.

Index