Package ‘filehashSQLite’

October 13, 2022

Version 0.2-6
Depends R (>= 4.0.0), filehash
Imports RSQLite, methods, DBI
Title Simple Key-Value Database Using SQLite
Author Roger D. Peng <rpeng@jhsph.edu>
Maintainer Roger D. Peng <rpeng@jhsph.edu>
Description Simple key-value database using SQLite as the back end.
License GPL (>= 2)
URL https://github.com/rdpeng/filehashsqlite
Encoding UTF-8
RoxygenNote 7.1.2
Suggests testthat (>= 3.0.0)
Config/testthat/edition 3
NeedsCompilation no
Repository CRAN
Date/Publication 2022-05-11 18:50:02 UTC

R topics documented:

- dbDelete, filehashSQLite, character-method ........................................ 2
- dbDisconnect, filehashSQLite-method ................................................. 2
- dbExists, filehashSQLite, character-method ....................................... 3
- dbFetch, filehashSQLite, character-method ......................................... 3
- dbInsert, filehashSQLite, character-method ....................................... 4
- dbList, filehashSQLite-method ......................................................... 4
- dbMultiFetch, filehashSQLite, character-method .................................. 5
- dbUnlink, filehashSQLite-method ..................................................... 5
- filehashSQLite .................................................................................. 6
- [, filehashSQLite, character, ANY, ANY-method ................................. 6

Index 7
dbDelete, filehashSQLite, character-method

*Delete Object*

**Description**

Delete an object from the database

**Usage**

```r
## S4 method for signature 'filehashSQLite, character'
dbDelete(db, key, ...)
```

**Arguments**

- `db` object of class "filehashSQLite"
- `key` character vector of key names
- `...` other arguments (not used)

---

dbDisconnect, filehashSQLite-method

*Disconnect from Database*

**Description**

Disconnect from Database

**Usage**

```r
## S4 method for signature 'filehashSQLite'
dbDisconnect(conn, ...)
```

**Arguments**

- `conn` database object
- `...` other arguments (not used)
**dbExists**, **filehashSQLite**, **character-method**

*Check Existence of Key*

**Description**

Check to see if a key is in the database

**Usage**

```r
## S4 method for signature 'filehashSQLite,character'
dbExists(db, key, ...)
```

**Arguments**

- `db` : object of class "filehashSQLite"
- `key` : character vector of key names
- `...` : other arguments (not used)

**dbFetch**, **filehashSQLite**, **character-method**

*Fetch Object*

**Description**

Retrieve the value associated with a specific key

**Usage**

```r
## S4 method for signature 'filehashSQLite,character'
dbFetch(db, key, ...)
```

**Arguments**

- `db` : object of class "filehashSQLite"
- `key` : character, key name
- `...` : other arguments (not used)
---

**dbInsert**, filehashSQLite-method

*Insert Object*

**Description**

Insert a key-value pair into a database

**Usage**

```
## S4 method for signature 'filehashSQLite,character'
dbInsert(db, key, value, ...)
```

**Arguments**

- `db`: object of class "filehashSQLite"
- `key`: character, key name
- `value`: R object
- `...`: other arguments (not used)

---

**dbList**, filehashSQLite-method

*List Keys*

**Description**

Return a character vector of all keys in the database

**Usage**

```
## S4 method for signature 'filehashSQLite'
dbList(db, ...)
```

**Arguments**

- `db`: object of class "filehashSQLite"
- `...`: other arguments (not used)
Fetch Multiple Objects

Description
Return (as a named list) the values associated with a vector of keys

Usage
```r
## S4 method for signature 'filehashSQLite,character'
dbMultiFetch(db, key, ...)
```

Arguments
- `db`: object of class "filehashSQLite"
- `key`: character vector of key names
- `...`: other arguments (not used)

Unlink Database

Description
Remove a database

Usage
```r
## S4 method for signature 'filehashSQLite'
dbUnlink(db, ...)
```

Arguments
- `db`: object of class "filehashSQLite"
- `...`: other arguments (not used)
**filehashSQLite**

*Filehash SQLite Class*

**Description**

Filehash SQLite Class

**Slots**

- `datafile` character, full path to the file in which the database should be stored
- `dbcon` Object of class “SQLiteConnection”, a SQLite connection
- `drv` ‘SQLite’ driver
- `name` character, the name of the database

**Note**

“filehashSQLite” databases have a "[" method that can be used to extract multiple elements in an efficient manner. The return value is a list with names equal to the keys passed to "["] . If there are keys passed to "[" that do not exist in the database, a warning is given.

The “SQLite” format for filehash uses an ASCII serialization of the data which could result in some rounding error for floating point numbers.

Note that if you use keys that are numbers coerced to character vectors, then you may have trouble with them being coerced to numeric. The SQLite database will see these key values and automatically convert them to numbers.

**[, filehashSQLite, character, ANY, ANY-method**

*Fetch Multiple Objects Operator*

**Description**

Return (as a named list) the values associated with a vector of keys

**Usage**

```r
## S4 method for signature 'filehashSQLite,character,ANY,ANY'
x[i, j, ..., drop = TRUE]
```

**Arguments**

- `x` object of class "filehashSQLite"
- `i` index
- `j` index
- `...` other arguments (not used)
- `drop` drop dimensions
Index

[,]filehashSQLite, character, ANY, ANY-method, 6

dbDelete, filehashSQLite, character-method, 2
dbDisconnect, filehashSQLite-method, 2
dbExists, filehashSQLite, character-method, 3
dbFetch, filehashSQLite, character-method, 3
dbInsert, filehashSQLite, character-method, 4
dbList, filehashSQLite-method, 4
dbMultiFetch, filehashSQLite, character-method, 5
dbUnlink, filehashSQLite-method, 5

filehashSQLite, 6
filehashSQLite-class (filehashSQLite), 6