Package 'neo4jshell'

October 13, 2022

Type Package

Title Querying and Managing 'Neo4J' Databases in 'R'

Version 0.1.2

Description Sends queries to a specified 'Neo4J' graph database, capturing results in a dataframe where appropriate.
Other useful functions for the importing and management of data on the 'Neo4J' server and basic local server admin.

License MIT + file LICENSE

Encoding UTF-8

Imports magrittr, ssh, sys, fs, R.utils

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

SystemRequirements neo4j - http://www.neo4j.com, cypher-shell - https://github.com/neo4j/cypher-shell/releases

VignetteBuilder knitr

NeedsCompilation no

Author Keith McNulty [cre, aut, cph] (<https://orcid.org/0000-0002-2332-1654>),
Liz Romero [ctb]

Maintainer Keith McNulty <keith.mcnulty@gmail.com>

Repository CRAN

Date/Publication 2022-04-11 09:50:02 UTC

R topics documented:

1

Index **9**

---

neo4j_import	*Imports a csv or a compressed file to Neo4J import folder.*

Description

Imports a csv or a compressed file to Neo4J import folder.

Usage

```
neo4j_import(
  local = FALSE,
  con = list(address = NULL, uid = NULL, pwd = NULL),
  source = NULL,
  import_dir = "import",
  unzip_path = "unzip",
  gunzip_path = "gunzip",
  tar_path = "tar"
)
```

Arguments

local	Logical indicating whether import is to a locally hosted or a remotely hosted server.
con	If remotely hosted server, list containing three objects: address, uid, pwd as character strings providing connection to the Neo4J server. uid and pwd must be for an account on the server with appropriate permissions.
source	Character string of local path to the csv, zip or tar.gz compressed csv file to be imported
import_dir	Character string of full path to the Neo4J import directory
unzip_path	Path to unzip on the local or remote server to be passed to the system command if necessary.
gunzip_path	Path to gunzip on the local or remote server to be passed to the system command following import if necessary.
tar_path	Path to tar on the local or remote server to be passed to the system command following import if necessary.

Value

System messages confirming success or error. zip or tar files will be removed after import and decompression.

Examples

```
# import zip to local import directory, with zip in the local system PATH variable
write.csv(mtcars, "mtcars.csv")
zip("mtcars.zip", "mtcars.csv")
fs::dir_create("import")
neo4j_import(local = TRUE, source = "mtcars.zip")
fs::file_delete("mtcars.zip")
fs::file_delete("mtcars.csv")
fs::dir_delete("import")
```

---

neo4j_query   *Execute a query string in Neo4J using cypher-shell and capture output*

Description

Execute a query string in Neo4J using cypher-shell and capture output

Usage

```
neo4j_query(
  con = list(address = NULL, uid = NULL, pwd = NULL),
  qry = NULL,
  shell_path = "cypher-shell",
  database = NULL,
  encryption = c("default", "true", "false")
)
```

Arguments

con	List containing three objects: bolt address, uid, pwd as character strings providing connection to the Neo4J server
qry	Character string of the query or queries to be sent to Neo4J. Read queries should be single queries.
shell_path	If cypher-shell is not in the PATH system variable, the full local path to cypher-shell executable.
database	The name of the database if other than the default database. (For multi-tenancy installations).
encryption	Passes encryption argument to cypher-shell if necessary. Older versions of cypher-shell may require 'true' or 'false' to be passed.

Value

A dataframe of results if the read query is successful. A text string if an error is encountered. Write queries will return a zero length response if successful. If multiple read queries were submitted, only the results of the final query will be returned.

Examples

```
# if neo4j exists, start the local server, give it a moment to fire up, and run a query
if (nzchar(Sys.which("neo4j"))) {
  neo4j_start()
  Sys.sleep(2)
  graph <- list(address = "bolt://localhost:7687", uid = "neo4j", pwd = "password")
  neo4j_query(con = graph, qry = "MATCH (n) RETURN (n)")
}
```

---

neo4j_restart   *Restart a local Neo4J database*

Description

Restart a local Neo4J database

Usage

```
neo4j_restart(neo4j_path = "neo4j")
```

Arguments

neo4j_path   Path to the Neo4J executable (usually in the bin directory of the Neo4J installa-
                  tion)

Value

System messages

Examples

```
# if neo4j exists, restart local graph with neo4j executable in the system PATH variable
if (nzchar(Sys.which("neo4j"))) {
  neo4j_restart()
}
```

---

neo4j_rmdir   *Remove subdirectory and all its contents from the Neo4J import direc-
                                tory*

Description

Remove subdirectory and all its contents from the Neo4J import directory

Usage

```
neo4j_rmdir(
  local = FALSE,
  con = list(address = NULL, uid = NULL, pwd = NULL),
  dir = NULL,
  import_dir = "import"
)
```

Arguments

local	Logical indicating whether import is to a locally hosted or remotely hosted server.
con	If remotely hosted server, list containing three objects: address, uid, pwd as character strings providing connection to the Neo4J server. uid and pwd must be for an account on the server with appropriate permissions.
dir	Character string of the Neo4J import subdirectory name to be deleted.
import_dir	Character string of path to the Neo4J import directory.

Value

A success message if successful. A error message otherwise.

Examples

```
# remove a subdirectory and all its contents from the local import directory
fs::dir_create("import/data")
fs::file_create("import/data/data.csv")
neo4j_rmdir(local = TRUE, dir = "data", import_dir = "import")
fs::dir_delete("import")
```

---

neo4j_rmfiles	*Remove files from the Neo4J import directory*

Description

Remove files from the Neo4J import directory

Usage

```
neo4j_rmfiles(
  local = FALSE,
  con = list(address = NULL, uid = NULL, pwd = NULL),
  files = NULL,
  import_dir = "import"
)
```

Arguments

`local`	Logical indicating whether import is to a locally hosted or remotely hosted server.
`con`	If remotely hosted server, list containing three objects: address, uid, pwd as character strings providing connection to the Neo4J server. uid and pwd must be for an account on the server with appropriate permissions.
`files`	Character vector of file names to be removed.
`import_dir`	Character string of path to the Neo4J import directory.

Value

A success message if successful. An error message otherwise.

Examples

```
# remove file from local import directory
fs::dir_create("import")
fs::file_create("import/data.csv")
neo4j_rmfiles(local = TRUE, files = "data.csv", import_dir = "import")
fs::dir_delete("import")
```

 `neo4j_start`   *Start a local Neo4J database*

Description

Start a local Neo4J database

Usage

```
neo4j_start(neo4j_path = "neo4j")
```

Arguments

`neo4j_path`	Path to the Neo4J executable (usually in the bin directory of the Neo4J installation)

Value

System messages

Examples

```
# if neo4j exists, start local graph on with neo4j executable in the system PATH variable
if (nzchar(Sys.which("neo4j"))) {
  neo4j_start()
}
```

---

neo4j_status	*Check status of a local Neo4J database*

Description

Check status of a local Neo4J database

Usage

```
neo4j_status(neo4j_path = "neo4j")
```

Arguments

neo4j_path   Path to the Neo4J executable (usually in the bin directory of the Neo4J installation)

Value

System messages

Examples

```
# if neo4j exists, check status local graph with neo4j executable in the system PATH variable
if (nzchar(Sys.which("neo4j"))) {
  neo4j_status()
}
```

---

neo4j_stop	*Stop a local Neo4J database*

Description

Stop a local Neo4J database

Usage

```
neo4j_stop(neo4j_path = "neo4j")
```

Arguments

neo4j_path   Path to the Neo4J executable (usually in the bin directory of the Neo4J installation)

Value

System messages

Examples

```
# if neo4j exists, stop local graph with neo4j executable in the system PATH variable
if (nzchar(Sys.which("neo4j"))) {
  neo4j_stop()
}
```

---

neo4j_wipe   *Wipe a complete local graph database in Neo4J*

Description

Wipe a complete local graph database in Neo4J

Usage

```
neo4j_wipe(database = NULL, data_path = NULL)
```

Arguments

database   Name of local graph database directory to wipe.

data_path   Path to the local Neo4J data directory

Value

Success or error message

Examples

```
# wipe database directory
fs::dir_create("data/databases/foo")
neo4j_wipe(database = "foo", data_path = "data")
fs::dir_delete("data")
```

Index