

Package ‘refreshr’

March 1, 2022

Type Package

Title Work with Refreshable Datasets that Update their Data Automatically

Description Connects dataframes/tables with a remote data source. Raw data downloaded from the data source can be further processed and transformed using data preparation code that is also baked into the dataframe/table. Refreshable dataframes can be shared easily (e.g. as R data files). Their users do not need to care about the inner workings of the data update mechanisms.

Version 0.1.0

Maintainer Joachim Zuckarelli <joachim@zuckarelli.de>

Depends R (>= 4.1.0)

License GPL-3

Imports stringr, crayon, lubridate, dplyr

Repository CRAN

BugReports <https://github.com/jsugarelli/refreshr/issues>

URL <https://github.com/jsugarelli/refreshr/>

Encoding UTF-8

ByteCompile true

RoxygenNote 7.1.1

NeedsCompilation no

Author Joachim Zuckarelli [aut, cre] (<<https://orcid.org/0000-0002-9280-3016>>)

Date/Publication 2022-03-01 08:30:02 UTC

R topics documented:

is.refreshr	2
make_refreshable	3
properties	4
refresh	5
refreshr	6
uptodate	7

is.refreshr	<i>Analysing refreshr objects</i>
-------------	-----------------------------------

Description

Checks if a dataframe/table is refreshable.

Usage

```
is.refreshr(df)
```

Arguments

df Dataframe/table to be checked.

Value

TRUE if the dataframe/table is of class refreshr (i.e. is of class "refreshr"), FALSE otherwise.

Examples

```
## Not run:

library(data.table)
library(dplyr)

# Load US unemployment rate from Bureau of Labor Statistics
data <- fread("https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData", sep="\t")

# Make refreshable and specify code for data preparation (filter raw data for
# the overall US employment rate) with # being a placeholder for the downloaded
# raw data
data_refresh <- make_refreshable(data,
  load_code = "data.table::fread(
    \"https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData\",
    sep=\"\t\")",
  prep_code = "filter(#[, series_id=\\\"LNS1400000\\\")")

# Save refreshable dataframe as RData file (e.g. to share dataset with coworkers or public)
save(data_refresh, file = "refresh.RData")

# Remove dataframe and reload it from file
rm(data_refresh)
load(file = "refresh.RData")

# Refresh the dataframe
data_refresh <- refresh(data_refresh)

# Show properties of refreshable dataframe
```

```

properties(data_refresh)

# Check if refreshable dataframe is up-to-date with the remote data source
uptodate(data_refresh)

## End(Not run)

```

make_refreshable *Making dataframes/tables refreshable*

Description

Makes a dataframe/table refreshable, i.e. connects it with a data source and specifies code that is applied to the raw data after the data has been loaded (optional).

Usage

```
make_refreshable(df, load_code, prep_code = NULL)
```

Arguments

df	The dataframe/table that is to be made refreshable
load_code	The code used to load the data from the data source. Please note that quotes need to be escaped (code\ <code\").< td=""> </code\").<>
prep_code	The code used to transform the raw data downloaded from the data source. The placeholder # can be used in this code to refer to the data downloaded from the data source.

Value

A dataframe/table of class refreshr that can be refreshed by calling refresh().

Examples

```

## Not run:

library(data.table)
library(dplyr)

# Load US unemployment rate from Bureau of Labor Statistics
data <- fread("https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData", sep="\t")

# Make refreshable and specify code for data preparation (filter raw data for
# the overall US employment rate) with # being a placeholder for the downloaded
# raw data
data_refresh <- make_refreshable(data,
  load_code = "data.table::fread(
    \"https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData\",
    sep=\"\t\")",

```

```

prep_code = "filter(#, series_id==\"LNS1400000\")"

# Save refreshable dataframe as RData file (e.g. to share dataset with coworkers or public)
save(data_refresh, file = "refresh.RData")

# Remove dataframe and reload it from file
rm(data_refresh)
load(file = "refresh.RData")

# Refresh the dataframe
data_refresh <- refresh(data_refresh)

# Show properties of refreshable dataframe
properties(data_refresh)

# Check if refreshable dataframe is up-to-date with the remote data source
uptodate(data_refresh)

## End(Not run)

```

properties

Analysing refreshr objects

Description

Checks if a dataframe/table is refreshable.

Usage

```
properties(df, property = NULL, silent = FALSE)
```

Arguments

<code>df</code>	Dataframe/table to be checked.
<code>property</code>	One-element Character vector describing the property that to be queried. Either "load" for the load code (the code that refreshes data from the data source), "prep" for the data preparation code (of any), "source" for the data source (which <code>properties()</code> tries to identify from the load code), "lastrefresh" (the date/timestamp of the last refresh of the dataframe/table). If no property is selected (<code>property == NULL</code> , the default) then all properties are included in the output to the screen.
<code>silent</code>	If silent the function will return (invisibly) the property defined by <code>property</code> without making any outputs on the screen. Default is FALSE.

Value

if `property == NULL`, i.e. all properties are queried, then `NULL` is returned. Otherwise `properties()` returns the value of the selected property.

Examples

```
## Not run:

library(data.table)
library(dplyr)

# Load US unemployment rate from Bureau of Labor Statistics
data <- fread("https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData", sep="\t")

# Make refreshable and specify code for data preparation (filter raw data for
# the overall US employment rate) with # being a placeholder for the downloaded
# raw data
data_refresh <- make_refreshable(data,
  load_code = "data.table::fread(
    \"https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData\",
    sep=\"\t\")",
  prep_code = "filter(, series_id=\"LNS1400000\")"
  #'
# Save refreshable dataframe as RData file (e.g. to share dataset with coworkers or public)
save(data_refresh, file = "refresh.RData")

# Remove dataframe and reload it from file
rm(data_refresh)
load(file = "refresh.RData")

# Refresh the dataframe
data_refresh <- refresh(data_refresh)

# Show properties of refreshable dataframe
properties(data_refresh)

# Check if refreshable dataframe is up-to-date with the remote data source
uptodate(data_refresh)

## End(Not run)
```

refresh

Working with refreshable dataframes/tables

Description

Refreshes a refreshable dataframes/table by downloading the data from the source and executing the data preparation code (if such code has been specified).

Usage

```
refresh(df, silent = FALSE)
```

Arguments

`df` The refreshed dataframe/table that is to be updated.
`silent` If TRUE then `refresh()` will not show any outputs on the screen.

Value

The refreshed dataframe/table with up-to-date data.

Examples

```
## Not run:

library(data.table)
library(dplyr)

# Load US unemployment rate from Bureau of Labor Statistics
data <- fread("https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData", sep="\t")

# Make refreshable and specify code for data preparation (filter raw data for
# the overall US employment rate) with # being a placeholder for the downloaded
# raw data
data_refresh <- make_refreshable(data,
  load_code = "data.table::fread(
    \"https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData\",
    sep=\"\t\")",
  prep_code = "filter(, series_id=\"LNS1400000\")")

# Save refreshable dataframe as RData file (e.g. to share dataset with coworkers or public)
save(data_refresh, file = "refresh.RData")

# Remove dataframe and reload it from file
rm(data_refresh)
load(file = "refresh.RData")

# Refresh the dataframe
data_refresh <- refresh(data_refresh)

# Show properties of refreshable dataframe
properties(data_refresh)

# Check if refreshable dataframe is up-to-date with the remote data source
uptodate(data_refresh)

## End(Not run)
```

Description

Create refreshable dataframes/tables that automatically pull in data from an (internet) data source and transform the data (if necessary) so that the user of your dataset does not have to worry about where to get the data from and how to update it.

Functions available:

- `make_refreshable()`: Makes a dataframe/table refreshable.
- `refresh()`: Refreshes a dataframe/table.
- `is.refreshr()`: Checks if a dataframe/table is set up as refreshable.
- `uptodate()`: Checks if a refreshable dataframe/table is up to date compared to the remote data source.
- `properties()`: Prints or returns the main properties of a refreshable dataframe/table.

uptodate

Updating dataframes/tables

Description

Checks if a refreshable dataframe/table is up-to-date with its data source.

Usage

```
uptodate(df)
```

Arguments

df Dataframe/table to be checked.

Details

Please note then `uptodate()` needs to download the data from the data source and process it according to the data preparation steps defined in the `prep` property of the refreshable dataframe/table in order to compare it to the current data of the refreshable dataframe/table. Depending on the amount of data and the complexity of the preparation steps this may take some time.

Value

TRUE if the dataframe/table properly reflects the state of its data source, FALSE otherwise.

Examples

```
## Not run:

library(data.table)
library(dplyr)

# Load US unemployment rate from Bureau of Labor Statistics
data <- fread("https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData", sep="\t")

# Make refreshable and specify code for data preparation (filter raw data for
# the overall US employment rate) with # being a placeholder for the downloaded
# raw data
data_refresh <- make_refreshable(data,
  load_code = "data.table::fread(
    \"https://download.bls.gov/pub/time.series/ln/ln.data.1.AllData\",
    sep=\"\t\")",
  prep_code = "filter(, series_id=\"LNS1400000\")")

# Save refreshable dataframe as RData file (e.g. to share dataset with coworkers or public)
save(data_refresh, file = "refresh.RData")

# Remove dataframe and reload it from file
rm(data_refresh)
load(file = "refresh.RData")

# Refresh the dataframe
data_refresh <- refresh(data_refresh)

# Show properties of refreshable dataframe
properties(data_refresh)

# Check if refreshable dataframe is up-to-date with the remote data source
uptodate(data_refresh)

## End(Not run)
```


Index

`is.refreshr`, [2](#), [7](#)

`make_refreshable`, [3](#), [7](#)

`properties`, [4](#), [7](#)

`refresh`, [5](#), [7](#)

`refreshr`, [6](#)

`uptodate`, [7](#), [7](#)