

Package ‘runonce’

October 7, 2025

Title Run Once and Save Result

Version 0.3.3

Description Package 'runonce' helps automating the saving of long-running code to help running the same code multiple times. If you run some long-running code once, it saves the result in a file on disk. Then, if the result already exists, i.e. if the code has already been run and its output has already been saved, it just reads the result from the stored file instead of running the code again.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports bigassertr, urltools

Suggests testthat (>= 2.1.0), covr

URL <https://github.com/privefl/runonce>,
<https://privefl.github.io/runonce/>

BugReports <https://github.com/privefl/runonce/issues>

NeedsCompilation no

Author Florian Privé [aut, cre]

Maintainer Florian Privé <florian.prive.21@gmail.com>

Repository CRAN

Date/Publication 2025-10-07 13:00:02 UTC

Contents

download_file	2
save_run	3
skip_run_if	4
Index	5

download_file

Download once

Description

Download file if does not exist yet.

Usage

```
download_file(
    url,
    dir = tempdir(),
    fname = url_basename(url),
    overwrite = FALSE,
    mode = "wb",
    timeout = max(300, getOption("timeout")),
    ...
)
```

Arguments

url	URL of file to be downloaded.
dir	Directory where to download the file.
fname	Base name of the downloaded file (dir will be prefixed).
overwrite	Whether to overwrite? Default is FALSE.
mode	See parameter of download.file() . Default of "wb" seems useful for Windows systems.
timeout	Number of seconds before timeout. Default is 300 (5 minutes), or <code>getOption("timeout")</code> when larger.
...	Arguments passed on to utils::download.file

method Method to be used for downloading files. Current download methods are "internal", "libcurl", "wget", "curl" and "wininet" (Windows only), and there is a value "auto": see 'Details' and 'Note'. The method can also be set through the option "download.file.method": see [options\(\)](#).

quiet If TRUE, suppress status messages (if any), and the progress bar.

cacheOK logical. Is a server-side cached value acceptable?

extra character vector of additional command-line arguments for the "wget" and "curl" methods.

headers named character vector of additional HTTP headers to use in HTTP[S] requests. It is ignored for non-HTTP[S] URLs. The User-Agent header taken from the HTTPUserAgent option (see [options](#)) is automatically used as the first header.

Value

Path to the downloaded (or existing) file.

Examples

```
download_file("https://github.com/privefl.png")
download_file("https://github.com/privefl.png")
download_file("https://github.com/privefl.png", overwrite = TRUE)
```

save_run	Cache result
----------	--------------

Description

Cache the result of code in an RDS file.

Usage

```
save_run(code, file, timing = TRUE)
```

Arguments

code	Code to run. Do not forget to wrap it with { }. Also, beware that it is your job to make sure your code and data has not changed . If this is the case, you need to remove the file storing the outdated result.
file	File path where the result is stored. Should have extension rds.
timing	Whether to print timing of running code? Default is TRUE.

Value

The evaluation of code the first time, the content of file otherwise.

Examples

```
# Prepare some temporary file
tmp <- tempfile(fileext = ".rds")

# Run once because result does not exist yet
save_run({ Sys.sleep(1); 1 }, file = tmp)

# Skip run because the result already exists
# (but still output how long it took the first time)
Sys.sleep(1); save_run({ Sys.sleep(1); 1 }, file = tmp)
```

skip_run_if	<i>Skip code</i>
-------------	------------------

Description

Skip code if all conditions are fulfilled. The code should not return anything.

Usage

```
skip_run_if(code, cond = NULL, files = NULL, timing = TRUE)
```

Arguments

code	Code to run. Do not forget to wrap it with { }.
cond	Condition to be fulfilled to skip running code. Default is NULL (not used). Should evaluate to either TRUE or FALSE otherwise.
files	Character vector of file path(s). Default is NULL (not used). This function checks if all these files exist, and if they all do exist, it skips running code.
timing	Whether to print timing of running code? Default is TRUE.

Value

NULL, invisibly.

Examples

```
# Prepare some temporary file
tmp <- tempfile(fileext = ".txt")

# Run once because file does not exist yet
skip_run_if({
  Sys.sleep(2)
  write.table(iris, tmp)
}, cond = file.exists(tmp))

# Skip run because `cond` is `TRUE`
skip_run_if({
  Sys.sleep(2)
  write.table(iris, tmp)
}, cond = file.exists(tmp))

# Skip run because file exists
skip_run_if({
  Sys.sleep(2)
  write.table(iris, tmp)
}, files = tmp)
```

Index

`download.file()`, [2](#)

`download_file`, [2](#)

`options`, [2](#)

`save_run`, [3](#)

`skip_run_if`, [4](#)

`utils::download.file`, [2](#)