

Package ‘resmush’

May 29, 2026

Title Optimize and Compress Image Files with 'reSmush.it'

Version 1.0.1

Description Optimize and compress local image files, directories and online images with the 'reSmush.it' API <<https://resmush.it/>>. Supports 'png', 'jpg/jpeg', 'gif', 'bmp' and 'tiff' files.

License MIT + file LICENSE

URL <https://dieghernan.github.io/resmush/>,
<https://github.com/dieghernan/resmush>

BugReports <https://github.com/dieghernan/resmush/issues>

Depends R (>= 4.1.0)

Imports cli, curl, httr2 (>= 1.0.0), tools, utils

Suggests grid, knitr, png, quarto, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder quarto

Config/Needs/website dieghernan/gitdevr, xfun, dplyr, tibble,
devtools, remotes, pak

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

X-schema.org-keywords r, compress-images, optimize-images, resmushit,
api, r-package, cran, cran-r

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-8457-4658>>)

Maintainer Diego Hernangómez <diego.hernangomezherrero@gmail.com>

Repository CRAN

Date/Publication 2026-05-29 10:50:02 UTC

Contents

resmush_dir	2
resmush_file	3
resmush_url	5

Index	8
--------------	----------

resmush_dir	<i>Optimize image files in directories</i>
-------------	--

Description

Optimize supported image files in one or more directories with the [reSmush.it API](#).

Usage

```
resmush_dir(
  dir,
  ext = "\\.(png|jpe?g|bmp|gif|tif)$",
  suffix = "_resmush",
  overwrite = FALSE,
  progress = TRUE,
  report = TRUE,
  recursive = FALSE,
  ...
)
```

Arguments

dir	A character vector of paths to local directories.
ext	A regex indicating the extensions of the files to optimize. The default captures all extensions supported by the API.
suffix	Character. Defaults to "_resmush". By default, optimized files are saved in the same directory with this suffix. For example, example.png becomes example_resmush.png. Values "", NA and NULL are equivalent to overwrite = TRUE.
overwrite	Logical. Should the files in dir be overwritten? If TRUE suffix is ignored.
progress	Logical. Should a progress bar be displayed?
report	Logical. Should a summary report be displayed in the console?
recursive	Logical. Should file search within dir be recursive? See also list.files() .
...	Arguments passed on to resmush_file
qlty	Integer between 0 and 100 indicating the optimization level. This only affects jpg files. For optimal results, use values above 90.
exif_preserve	Logical. Should Exif metadata be preserved? The default is FALSE, which removes it.

Value

Writes optimized files to disk when the API call is successful. Invisibly returns a data frame summarizing the process.

See Also

[reSmush.it API](#) documentation.

See [resmush_clean_dir\(\)](#) to clean a directory of previous runs.

Other functions for optimizing: [resmush_file\(\)](#), [resmush_url\(\)](#)

Examples

```
# Copy the example directory.
example_dir <- system.file("extimg", package = "resmush")
temp_dir <- tempdir()
file.copy(example_dir, temp_dir, recursive = TRUE)

# Create the destination folder path.
dest_folder <- file.path(tempdir(), "extimg")

# Non-recursive.
resmush_dir(dest_folder)
resmush_clean_dir(dest_folder)

# Recursive.
summary <- resmush_dir(dest_folder, recursive = TRUE)

# Return the same information in the invisible data frame.
summary[, -c(1, 2)]

# Display the `png` output.
if (require("png", quietly = TRUE)) {
  a_png <- grepl("png$", summary$dest_img)
  my_png <- png::readPNG(summary[a_png, ]$dest_img[2])
  grid::grid.raster(my_png)
}

# Clean up the example files.
unlink(dest_folder, force = TRUE, recursive = TRUE)
```

resmush_file

Optimize local image files

Description

Optimize one or more local image files with the [reSmush.it API](#).

Usage

```
resmush_file(
  file,
  suffix = "_resmush",
  overwrite = FALSE,
  progress = TRUE,
  report = TRUE,
  qlty = 92,
  exif_preserve = FALSE
)
```

Arguments

file	Path or paths to local image files. The API can optimize png, jpg/jpeg, gif, bmp and tiff files.
suffix	Character. Defaults to "_resmush". By default, the optimized file is saved in the same directory as file with this suffix. For example, example.png becomes example_resmush.png. Values "", NA and NULL are equivalent to overwrite = TRUE.
overwrite	Logical. Should the file in file be overwritten? If TRUE suffix is ignored.
progress	Logical. Should a progress bar be displayed?
report	Logical. Should a summary report be displayed in the console?
qlty	Integer between 0 and 100 indicating the optimization level. This only affects jpg files. For optimal results, use values above 90.
exif_preserve	Logical. Should Exif metadata be preserved? The default is FALSE, which removes it.

Value

Writes optimized files to disk when the API call is successful. Invisibly returns a data frame summarizing the process. With report = TRUE, a summary report is also displayed in the console.

See Also

reSmush.it API documentation.

See [resmush_clean_dir\(\)](#) to clean a directory of previous runs.

Other functions for optimizing: [resmush_dir\(\)](#), [resmush_url\(\)](#)

Examples

```
png_file <- system.file("extimg/example.png", package = "resmush")

# Copy to a temporary file for this example.
tmp_png <- tempfile(fileext = ".png")
```

```
file.copy(png_file, tmp_png, overwrite = TRUE)

resmush_file(tmp_png)

# Several paths.
jpg_file <- system.file("extimg/example.jpg", package = "resmush")
tmp_jpg <- tempfile(fileext = ".jpg")

file.copy(jpg_file, tmp_jpg, overwrite = TRUE)

# Display a summary in the console.
summary <- resmush_file(c(tmp_png, tmp_jpg))

# The invisible data frame contains the same information.
summary

# Display the `png` output.
if (require("png", quietly = TRUE)) {
  my_png <- png::readPNG(summary$dest_img[1])
  grid::grid.raster(my_png)
}

# Use with `jpg` files and parameters.
resmush_file(tmp_jpg)
resmush_file(tmp_jpg, qlty = 10)
```

resmush_url

Optimize online image files

Description

Optimize and download one or more online image files with the [reSmush.it API](https://reSmush.it).

Usage

```
resmush_url(
  url,
  outfile = file.path(tempdir(), basename(url)),
  overwrite = FALSE,
  progress = TRUE,
  report = TRUE,
  qlty = 92,
  exif_preserve = FALSE
)
```

Arguments

url	URL or vector of URLs pointing to hosted image files. The API can optimize png, jpg/jpeg, gif, bmp and tiff files.
outfile	Path or paths where optimized files are stored on disk. By default, temporary files are created with <code>tempfile()</code> and the same <code>basename()</code> as the file provided in url. outfile must have the same length as url.
overwrite	Logical. Should outfile be overwritten if it already exists? If FALSE and outfile exists, a copy is created with a numerical suffix, such as <code><outfile>_01.png</code> .
progress	Logical. Should a progress bar be displayed?
report	Logical. Should a summary report be displayed in the console?
qlty	Integer between 0 and 100 indicating the optimization level. This only affects jpg files. For optimal results, use values above 90.
exif_preserve	Logical. Should Exif metadata be preserved? The default is FALSE, which removes it.

Value

Writes optimized files to disk when the API call is successful. Invisibly returns a data frame summarizing the process. If any value in outfile is duplicated, `resmush_url()` renames the outputs with suffixes such as `_01` and `_02`.

See Also

reSmush.it API documentation.

Other functions for optimizing: `resmush_dir()`, `resmush_file()`

Examples

```
# Base URL.
base_url <- "https://raw.githubusercontent.com/dieghernan/resmush/main/inst/"

png_url <- paste0(base_url, "/extimg/example.png")
resmush_url(png_url)

# Several URLs.
jpg_url <- paste0(base_url, "/extimg/example.jpg")

summary <- resmush_url(c(png_url, jpg_url))

# The invisible data frame contains a summary of the process.
summary

# Display the `png` output.
if (require("png", quietly = TRUE)) {
  my_png <- png::readPNG(summary$dest_img[1])
  grid::grid.raster(my_png)
```

```
}  
  
# Use with `jpg` files and parameters.  
resmush_url(jpg_url)  
resmush_url(jpg_url, qlty = 10)
```

Index

* **optimize**

- resmush_dir, 2
- resmush_file, 3
- resmush_url, 5

basename(), 6

list.files(), 2

regex, 2

resmush_clean_dir(), 3, 4

resmush_dir, 2

resmush_dir(), 4, 6

resmush_file, 2, 3

resmush_file(), 3, 6

resmush_url, 5

resmush_url(), 3, 4

tempfile(), 6