Package 'resmush'

August 25, 2026

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

Version 1.0.3

Description Optimize and compress local and online image files with the 'reSmush.it' 'API' <https://resmush.it/api/>. Process individual files or entire directories. The 'API' is free for personal use, accepts files smaller than 5 'MB' and supports 'PNG', '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, testthat (>= 3.3.0), withr

VignetteBuilder quarto

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

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.0

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 2026-08-25 11:20:02 UTC

Contents

resmush_clean_dir 2
resmush_dir 3
resmush_file 5
resmush_url 6

Index **9**

`resmush_clean_dir` *Remove output files from directories*

Description

Use with caution. Remove files that match `suffix` from one or more directories. This function cleans output files created by `resmush_file()` or `resmush_dir()`.

Usage

```
resmush_clean_dir(dir, suffix = "_resmush", recursive = FALSE)
```

Arguments

`dir`	A character vector of directory paths. See the `path` argument of `base::list.files()`.
`suffix`	A character string containing the suffix pattern used to identify files. The value is interpreted as a regular expression. The default is `"_resmush"`, which is also used by `resmush_file()`.
`recursive`	Logical. Should the file search recurse into directories?

Value

An `base::invisible()` `NULL`. Messages list the files selected for removal.

See Also

`resmush_file()` and `resmush_dir()` create the suffixed output files that this function removes.

Examples

```
# Create a temporary file with a suffix to remove.
png_file <- system.file("extimg/example.png", package = "resmush")
suffix <- "_would_be_removed"
tmp_png <- file.path(
  tempdir(),
  paste0("example", suffix, ".png")
)

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

```
file.exists(tmp_png)

# Run with the default suffix. This should not remove the file.
resmush_clean_dir(tempdir())

file.exists(tmp_png)

# Use the matching suffix to remove the file.
resmush_clean_dir(tempdir(), suffix = suffix)

file.exists(tmp_png)
```

`resmush_dir` *Optimize image files in directories*

Description

Optimize supported image files in one or more directories with the reSmush.it API. The API is free for personal use and accepts files smaller than 5 MB.

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` matching file extensions to optimize. The default matches lowercase `.png`, `.jpg`, `.jpeg`, `.gif`, `.bmp` and `.tif` extensions.
`suffix`	A character string inserted before each output file extension. The default is `"_resmush"`. Therefore, `example.png` becomes `example_resmush.png`. Values `""`, `NA` and `NULL` are equivalent to `overwrite = TRUE`.
`overwrite`	Logical. Should the input files 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 the file search within each directory in `dir` be recursive? See `base::list.files()`.

`...` Arguments passed on to `resmush_file`

`qlty` An integer between `0` and `100` indicating the JPEG quality level. For best results, use values above `90`. This argument only affects JPEG files.

`exif_preserve` Logical. Should EXIF metadata be preserved? The default is `FALSE`, which removes it.

Value

An invisibly returned data frame with one row per result and columns containing source and destination paths, formatted and raw file sizes, compression ratios and status notes. Returns `NULL` if no result is available. Successful API calls also write the optimized files to disk. If `report = TRUE`, a summary is displayed in the console.

See Also

`resmush_clean_dir()` removes output files created by previous runs. The reSmush.it API documentation describes the external service.

Other image optimization functions: `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 directory path.
dest_folder <- file.path(tempdir(), "extimg")

# Optimize files non-recursively.
resmush_dir(dest_folder)
resmush_clean_dir(dest_folder)

# Optimize files recursively.
summary <- resmush_dir(dest_folder, recursive = TRUE)

# Inspect the returned optimization summary.
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. The API is free for personal use and accepts files smaller than 5 MB.

Usage

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

Arguments

`file`	A character vector of paths to local image files. The API can optimize PNG, JPEG, GIF, BMP and TIFF files.
`suffix`	A character string inserted before each output file extension. The default is `"_resmush"`. Therefore, `example.png` becomes `example_resmush.png`. Values `""`, `NA` and `NULL` are equivalent to `overwrite = TRUE`.
`overwrite`	Logical. Should the input files 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`	An integer between `0` and `100` indicating the JPEG quality level. For best results, use values above `90`. This argument only affects JPEG files.
`exif_preserve`	Logical. Should EXIF metadata be preserved? The default is `FALSE`, which removes it.

Value

An invisibly returned data frame with one row per result and columns containing source and destination paths, formatted and raw file sizes, compression ratios and status notes. Returns `NULL` if no result is available. Successful API calls also write the optimized files to disk. If `report = TRUE`, a summary is displayed in the console.

See Also

`resmush_clean_dir()` removes output files created by previous runs. The reSmush.it API documentation describes the external service.

Other image optimization functions: `resmush_dir()`, `resmush_url()`

Examples

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

# Copy the example PNG to a temporary file.
tmp_png <- tempfile(fileext = ".png")

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

resmush_file(tmp_png)

# Optimize multiple files.
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))

# Inspect the returned optimization summary.
summary

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

# Adjust the JPEG quality level.
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. The API is free for personal use and accepts files smaller than 5 MB.

Usage

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

Arguments

`url`	A character vector of URLs pointing to hosted image files. The API can optimize PNG, JPEG, GIF, BMP and TIFF files.
`outfile`	A character vector of paths where optimized files are stored. By default, files are created in `base::tempdir()` with the same `base::basename()` as each file in `url`. `outfile` must have the same length as `url`.
`overwrite`	Logical. Should existing files in `outfile` be overwritten? If `FALSE`, existing paths are made unique with a numeric suffix, such as `example_01.png`.
`progress`	Logical. Should a progress bar be displayed?
`report`	Logical. Should a summary report be displayed in the console?
`qlty`	An integer between `0` and `100` indicating the JPEG quality level. For best results, use values above `90`. This argument only affects JPEG files.
`exif_preserve`	Logical. Should EXIF metadata be preserved? The default is `FALSE`, which removes it.

Details

If `outfile` contains duplicate paths, `resmush_url()` makes them unique with suffixes such as `_01` and `_02` unless `overwrite = TRUE`.

Value

An invisibly returned data frame with one row per result and columns containing source and destination paths, formatted and raw file sizes, compression ratios and status notes. Returns `NULL` if no result is available. Successful API calls also write the optimized files to disk. If `report = TRUE`, a summary is displayed in the console.

See Also

The reSmush.it API documentation describes the external service.

Other image optimization functions: `resmush_dir()`, `resmush_file()`

Examples

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

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

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

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

# Inspect the returned optimization summary.
summary

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

# Adjust the JPEG quality level.
resmush_url(jpg_url)
resmush_url(jpg_url, qlty = 10)
```

Index

∗ **helpers**
resmush_clean_dir, 2
∗ **optimize**
resmush_dir, 3
resmush_file, 5
resmush_url, 6

base::basename(), *7*
base::invisible(), *2*
base::list.files(), *2*, *4*
base::tempdir(), *7*

regex, *3*
resmush_clean_dir, 2
resmush_clean_dir(), *4*, *6*
resmush_dir, 3
resmush_dir(), *2*, *6*, *7*
resmush_file, *4*, 5
resmush_file(), *2*, *4*, *7*
resmush_url, 6
resmush_url(), *4*, *6*