

# Package: tinieR (via r-universe)

October 15, 2024

**Title** Shrink Image Filesizes with TinyPNG.com API

**Version** 0.5.0

**Description** Shrink image filesizes with the TinyPNG API  
<<https://tinypng.com>>. From the TinyPNG website: ``TinyPNG uses smart lossy compression techniques to reduce the file size of your files. By selectively decreasing the number of colors in the image, fewer bytes are required to store the data. The effect is nearly invisible but it makes a very large difference in file size!'' TinieR works with .png and .jpg/.jpeg files, and can return the new image filepath to enable embedding in other image workflows/functions.

**License** MIT + file LICENSE

**URL** <https://github.com/jmablog/tinieR>

**BugReports** <https://github.com/jmablog/tinieR/issues>

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Imports** fs (>= 1.4.2), glue (>= 1.4.1), httr (>= 1.4.1), jsonlite (>= 1.7.0), png, jpeg, rprojroot, yaml, cli, grDevices, ggplot2

**Suggests** testthat, knitr, rmarkdown, ragg, withr

**VignetteBuilder** knitr

**Repository** <https://jmablog.r-universe.dev>

**RemoteUrl** <https://github.com/jmablog/tinieR>

**RemoteRef** HEAD

**RemoteSha** fa269a3df404f93d5f9e0b38ea3cd8897f34abab

## Contents

petit_plot . . . . .	2
tinify . . . . .	4
tinify_defaults . . . . .	7
tinify_key . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

petit_plot	<i>Save a plot to a file and automatically shrink it</i>
------------	--

---

## Description

Save a plot to a file and automatically shrink it with `tinify()`.

## Usage

```
petit_plot(
  filename = "plot",
  path = NULL,
  device = "png",
  ragg = FALSE,
  keep_large = FALSE,
  suffix,
  quiet,
  return_path,
  key = NULL,
  ...
)

petit_ggplot(
  filename = "plot",
  path = NULL,
  plot = ggplot2::last_plot(),
  device = "png",
  keep_large = FALSE,
  suffix,
  quiet,
  return_path,
  key = NULL,
  ...
)
```

## Arguments

filename	String, required. The name to give the output image file. Do not include a file extension.
----------	--

path	String, optional. If NULL, defaults to the current working directory. The path to save the image file into, relative to the current working directory. Do not include the final trailing path separator.
device	String, optional. Defaults to "png". One of "png" or "jpg", to choose the output image file type. TinyPNG only supports png or jpg image types.
ragg	Boolean, optional. Defaults to FALSE. Whether to use the <b>ragg</b> package as the device backend to generate the image. Will error if <b>ragg</b> is not installed, but does not need to be loaded.
keep_large	Boolean, optional. Defaults to FALSE. Whether to keep the unshrunk original image file alongside the tiny version, or just keep the shrunk file.
suffix	String, optional. If keep_large is TRUE, the suffix to add to the shrunk file. Ignored if keep_large is FALSE.
quiet	Boolean, optional. If set to TRUE, tinify() displays no information messages as it shrinks files.
return_path	String or NULL, optional. One of "proj", "rel", "abs", or "all". If "proj", will return the file path of the newly tinified image file relative to the Rstudio project directory (looking for an .Rproj file). If no project can be identified, returns NA. If "rel", will return the file path of the newly tinified image file, relative to the <b>current</b> working directory at the time tinify() is called. If "abs", will return the absolute file path of the newly tinified image file. If "all", will return a named list with all file paths. If NULL (the default), no file path is returned.
key	String, optional. A string containing your TinyPNG API key. Not required if the API key is set using tinify_api(). Any other key provided as an argument will override the key set by tinify_api().
...	Additional plot options, passed directly to either png()/ragg::agg_png(), jpeg()/ragg::agg_jpeg(), or ggplot2::ggsave() depending on method used.
plot	Object, optional. The plot object to export. Defaults to the last plot modified or created (using ggplot2::last_plot()) if not provided.

## Details

These are convenience functions to wrap saving either a base R plot with png() or jpeg() devices, or a **ggplot** object with ggplot2::ggsave(), before passing the resulting image file directly to tinify(). Can also make use of the **ragg** package for base plots if installed.

## Value

If return\_path = "proj", return\_path = "rel", or return\_path = "abs", a string with the project, relative, or absolute path to the newly tinified image file. If no project can be identified for return\_path = "proj", returns NA. If return\_path = "all", a named list with all file paths included as \$project, \$relative, and \$absolute respectively. If return\_path = NULL, no return value.

## Using with base R plots

To save and shrink a base R plot, print the plot and call petit\_plot() immediately after. Under the hood, this uses recordPlot() to capture and replay the last plot created within the chosen device with the applied options:

```
plot(mtcars$mpg, mtcars$drat)

petit_plot(filename = "mtcars")
```

### Using with ggplot

To save and shrink a **ggplot**, either create, modify, or show the plot and call `petit_plot()` immediately after, in a similar process to base R plots:

```
ggplot(data = palmerpenguins::penguins,
       aes(flipper_length_mm, body_mass_g)) +
geom_point(aes(color = species))

petit_plot(filename = "penguins")
```

Or use `petit_ggplot()` to capture specifically the last `ggplot` created or modified:

```
ggplot(data = palmerpenguins::penguins,
       aes(flipper_length_mm, body_mass_g)) +
geom_point(aes(color = species))

petit_ggplot(filename = "penguins")
```

Or provide the plot object explicitly to `petit_ggplot()` with `plot`:

```
p <- ggplot(data = palmerpenguins::penguins,
           aes(flipper_length_mm, body_mass_g)) +
  geom_point(aes(color = species))

petit_ggplot(filename = "penguins", plot = p)
```

### See Also

[tinify\(\)](#)

[tinify\\_key\(\)](#) to set an API key globally

[tinify\\_defaults\(\)](#) to set default arguments that will be used if not provided explicitly

---

`tinify`

*Shrink image files with TinyPNG*

---

### Description

Shrink an image's (PNG or JPG) filesize with the TinyPNG API.

### Usage

```
tinify(file, overwrite, suffix, quiet, return_path, resize, key = NULL)
```

## Arguments

file	String, required. A string detailing the path to the file you wish to shrink, relative to the current working directory or as an absolute file path. Can include sub-directories and must include the file extension (.png or .jpg/.jpeg only).
overwrite	Boolean, defaults to FALSE. By default, tinify will create a new file with the suffix '_tiny' and preserve the original file. Set TRUE to instead overwrite the original file, with the same file name.
suffix	String, defaults to "_tiny". By default, tinify will create a new file with the suffix '_tiny' and preserve the original file. Provide a new character string here to change the suffix from '_tiny' to your own choice. Empty strings ("") are not accepted. suffix is ignored when overwrite is set to TRUE.
quiet	Boolean, defaults to FALSE. By default, tinify provides details on file names, amount of file size reduction (% and Kb), and the number of TinyPNG API calls made this month. If set to TRUE, tinify displays no messages as it shrinks files.
return_path	String or NULL, optional. One of "proj", "rel", "abs", or "all". If "proj", will return the file path of the newly tinified image file relative to the Rstudio project directory (looking for an .Rproj file). If no project can be identified, returns NA. If "rel", will return the file path of the newly tinified image file, relative to the <b>current</b> working directory at the time tinify() is called. If "abs", will return the absolute file path of the newly tinified image file. If "all", will return a named list with all file paths. If NULL (the default), no file path is returned.
resize	Named list or NULL, optional. A named list with the elements method as a string, and width and/or height as numerics. Please note you can only reduce an image's dimensions and make an image smaller with TinyPNG API, not make an image larger. Method must be set to one of "scale", "fit", "cover", or "thumb". If using "scale", you only need to provide width OR height, not both. If using any other method, you must supply both a width AND height. See <a href="https://tinypng.com/developers/reference#resizing-images">https://tinypng.com/developers/reference#resizing-images</a> and the examples for more. If NULL (the default), no resizing takes place.
key	String, optional. A string containing your TinyPNG API key. Not required if the API key is set using tinify_api(). If an API key is provided with tinify_api(), any other key provided in the function call will override the key set by tinify_api().

## Value

If return\_path = "proj", return\_path = "rel", or return\_path = "abs", a string with the project, relative, or absolute path to the newly tinified image file. If no project can be identified for return\_path = "proj", returns NA. If return\_path = "all", a named list with all file paths included as \$project, \$relative, and \$absolute respectively. If return\_path = NULL, no return value.

## Note

If any argument is provided to tinify() when called, it will overwrite the default option set by tinify\_defaults().

## TinyPNG API

You can get a TinyPNG API key from <https://tinypng.com/developers>. TinyPNG is smart enough to know when you are uploading the same file again, and so will not count repeat calls of `tinify()` on the same image file against your monthly API usage limit. This can be useful if, for example, you are using `tinify()` in an RMarkdown document as it won't count against your API usage every time you knit your document. But, be aware that use of `resize` also counts as an additional API call, as the image is first reduced in filesize, then a second API call is made to resize the newly tinified file.

## See Also

`tinify_key()` to set an API key globally so it does not need to be provided with every call of `tinify()`

`tinify_defaults()` to set default arguments so they do not need to be provided with every call of `tinify()`

## Examples

```
## Not run:
# Shrink a PNG file

img <- system.file("extdata", "example.png", package = "tinieR")

tinify(img)

# Also works with JPEG/JPG files

img_jpg <- system.file("extdata", "example.jpg", package = "tinieR")

tinify(img_jpg)

# Return absolute path to newly shrunk file:

shrunk_img <- tinify(img, return_path = "abs")

# Suppress messages detailing file reduction amount:

tinify(img, quiet = TRUE)

# Overwrite original file in place:

tinify(img, overwrite = TRUE)

# Change suffix on new file:

tinify(img, suffix = "_small")

# Resize an image with the method "scale", only providing a width:

tinify(img, resize = list(method = "scale", width = 300))
```

```
# Or resize an image with any other method by providing both width and height:

tinify(img, resize = list(method = "cover", width = 300, height = 150))

# Overwrite a global API key set in tinify_api():

tinify(img, key = "NEW-API-KEY-HERE")

# You can combine any of the above:

tinify(img,
        overwrite = TRUE,
        quiet = TRUE,
        return_path = "rel")

# Plays nice with the pipe:

img %>% tinify()

# And with purrr::map for multiple files:

imgs <- c("example.png", "example2.png")

purrr::map(imgs, ~tinify(.x))

# An example method for shrinking an entire directory:

imgs_dir <- fs::dir_ls("imgs", glob = "*.png")

purrr::map(imgs_dir, ~tinify(.x, overwrite = TRUE, quiet = TRUE))

## End(Not run)
```

---

<code>tinify_defaults</code>	<i>Set defaults for tinify() function</i>
------------------------------	---

---

## Description

Set some default options for `tinify()` in the global environment, so it is no longer necessary to explicitly provide each argument with every call of `tinify()`.

If called without any arguments, `tinify_defaults()` will print the current default options set to the console.

## Usage

```
tinify_defaults(overwrite, suffix, quiet, return_path, resize)
```

## Arguments

overwrite	Boolean, defaults to FALSE. By default, tinify will create a new file with the suffix '_tiny' and preserve the original file. Set TRUE to instead overwrite the original file, with the same file name.
suffix	String, defaults to "_tiny". By default, tinify will create a new file with the suffix '_tiny' and preserve the original file. Provide a new character string here to change the suffix from '_tiny' to your own choice. Empty strings ("") are not accepted. suffix is ignored when overwrite is set to TRUE.
quiet	Boolean, defaults to FALSE. By default, tinify provides details on file names, amount of file size reduction (% and Kb), and the number of TinyPNG API calls made this month. If set to TRUE, tinify displays no messages as it shrinks files.
return_path	String or NULL, optional. One of "proj", "rel", "abs", or "all". If "proj", will return the file path of the newly tinified image file relative to the Rstudio project directory (looking for an .Rproj file). If no project can be identified, returns NA. If "rel", will return the file path of the newly tinified image file, relative to the <b>current</b> working directory at the time tinify() is called. If "abs", will return the absolute file path of the newly tinified image file. If "all", will return a named list with all file paths. Set to NULL to stop returning a file path.
resize	Named list or NULL, optional. A named list with the elements method as a string, and width and/or height as numerics. Please note you can only reduce an image's dimensions and make an image smaller with TinyPNG API, not make an image larger. Method must be set to one of "scale", "fit", "cover", or "thumb". If using "scale", you only need to provide width OR height, not both. If using any other method, you must supply both a width AND height. See <a href="https://tinypng.com/developers/reference#resizing-images">https://tinypng.com/developers/reference#resizing-images</a> and the examples for more. Set to NULL to stop resizing.

## See Also

[tinify\(\)](#) to shrink image filesizes  
[tinify\\_key\(\)](#) to set a default TinyPNG.com API key

## Examples

```
## Not run:

tinify_defaults(quiet = TRUE, suffix = "_small")

# show current defaults set

tinify_defaults()
#> Tinify 'overwrite' set to: FALSE
#> Tinify 'suffix' set to: "_small"
#> Tinify 'quiet' set to: FALSE
#> Tinify 'return_path' set to: No return
#> Tinify 'resize' set to: No resize

## End(Not run)
```



---

tinify_key	<i>Set TinyPNG.com API Key</i>
------------	--------------------------------

---

**Description**

Set your TinyPNG API key in the system environment, so it is no longer necessary to explicitly provide an API key with every call of `tinify()`.

**Usage**

```
tinify_key(key)
```

**Arguments**

key	A string containing your TinyPNG.com API key.
-----	---

**Details**

You can get a TinyPNG API key from <https://tinypng.com/developers>.

**See Also**

[tinify\(\)](#) to shrink image file sizes

[tinify\\_defaults\(\)](#) to set default arguments so they do not need to be provided with every call of `tinify()`

**Examples**

```
## Not run:  
tinify_key("YOUR-API-KEY-HERE")  
  
## End(Not run)
```

# Index

`petit_ggplot (petit_plot)`, [2](#)  
`petit_plot`, [2](#)

`tinify`, [4](#)  
`tinify()`, [4](#), [8](#), [9](#)  
`tinify_defaults`, [7](#)  
`tinify_defaults()`, [4](#), [6](#), [9](#)  
`tinify_key`, [9](#)  
`tinify_key()`, [4](#), [6](#), [8](#)