

Package ‘downloadthis’

August 16, 2022

Title Implement Download Buttons in 'rmarkdown'

Version 0.3.2

Description Implement download buttons in HTML output from 'rmarkdown' without the need for 'runtime:shiny'.

License MIT + file LICENSE

URL <https://github.com/fmmattioni/downloadthis>

BugReports <https://github.com/fmmattioni/downloadthis/issues>

Encoding UTF-8

Imports fs, readr, writexl, bsplus, mime, htmltools, magrittr, zip, ggplot2, base64enc

RoxygenNote 7.2.1

Suggests knitr, rmarkdown, testthat (>= 3.0.1), spelling, covr

VignetteBuilder knitr

Language en-US

Config/testthat/edition 3

Config/testthat/parallel true

NeedsCompilation no

Author Felipe Mattioni Maturana [aut, cre]
(<<https://orcid.org/0000-0002-4221-6104>>),
John Coene [ctb]

Maintainer Felipe Mattioni Maturana <felipe.mattioni@med.uni-tuebingen.de>

Repository CRAN

Date/Publication 2022-08-16 14:10:06 UTC

R topics documented:

download_dir	2
download_file	3
download_link	4
download_this	5

Index**8**

download_dir	<i>Download a directory</i>
--------------	-----------------------------

Description

The path is converted into a .zip file.

Usage

```
download_dir(
  path,
  output_name,
  button_label = "Download data",
  button_type = c("default", "primary", "success", "info", "warning", "danger"),
  has_icon = TRUE,
  icon = "fa fa-save",
  self_contained = FALSE,
  ...
)
```

Arguments

path	Path to directory.
output_name	Name of of the output file.
button_label	Character (HTML), button label
button_type	Character, one of the standard Bootstrap types
has_icon	Specify whether to include fontawesome icons in the button label
icon	Fontawesome tag e.g.: "fa fa-save"
self_contained	A boolean to specify whether your HTML output is self-contained. Default to FALSE.
...	attributes (named arguments) and children (unnamed arguments) of the button, passed to <code>htmltools::tag()</code> .

Value

`htmltools::tag`, `<button>`

Examples

```
## Not run:
## Directory path as an example
download_dir(
  path = system.file("assets", package = "downloadthis"),
  output_name = "example dir",
  button_label = "Download directory",
```

```

    button_type = "success",
    has_icon = TRUE,
    icon = "fa fa-save",
    self_contained = FALSE
  )

  ## End(Not run)

```

 download_file

Download a local file or multiple files

Description

In case multiple files are chosen, the files will be converted to a .zip file.

Usage

```

download_file(
  path,
  output_name,
  button_label = "Download data",
  button_type = c("default", "primary", "success", "info", "warning", "danger"),
  has_icon = TRUE,
  icon = "fa fa-save",
  self_contained = FALSE,
  ...
)

```

Arguments

path	Path to the file(s). If multiple files are chosen, a vector must be passed to this argument.
output_name	Name of of the output file. If not specified, it will take the source file's name if one file is specified. In case of multiple files, the output_name must be specified.
button_label	Character (HTML), button label
button_type	Character, one of the standard Bootstrap types
has_icon	Specify whether to include fontawesome icons in the button label
icon	Fontawesome tag e.g.: "fa fa-save"
self_contained	A boolean to specify whether your HTML output is self-contained. Default to FALSE.
...	attributes (named arguments) and children (unnamed arguments) of the button, passed to <code>htmltools::tag()</code> .

Value

`htmltools::tag`, `<button>`

Examples

```

## Not run:
## One file example
download_file(
  path = system.file("assets/css/all.min.css", package = "downloadthis"),
  output_name = "CSS file from downloadthis",
  button_label = "Download css file",
  button_type = "danger",
  has_icon = TRUE,
  icon = "fa fa-save",
  self_contained = FALSE
)

## Multiple files example
path_files <- list.files(
  path = system.file("assets/css", package = "downloadthis"),
  full.names = TRUE
)

download_file(
  path = path_files,
  output_name = "Files from downloadthis",
  button_label = "Download files",
  button_type = "danger",
  has_icon = TRUE,
  icon = "fa fa-save",
  self_contained = FALSE
)

## End(Not run)

```

download_link

Download file from a web address

Description

It associates your download button to a web link. This might be an alternative when your file is too big for being handled by `download_this()`.

Usage

```

download_link(
  link,
  button_label = "Download data",
  button_type = c("default", "primary", "success", "info", "warning", "danger"),
  has_icon = TRUE,
  icon = "fa fa-save",
  self_contained = FALSE,
  ...
)

```

Arguments

link	A web address for downloading the file.
button_label	Character (HTML), button label
button_type	Character, one of the standard Bootstrap types
has_icon	Specify whether to include fontawesome icons in the button label
icon	Fontawesome tag e.g.: "fa fa-save"
self_contained	A boolean to specify whether your HTML output is self-contained. Default to FALSE.
...	attributes (named arguments) and children (unnamed arguments) of the button, passed to <code>htmltools::tag()</code> .

Value

`htmltools::tag`, `<button>`

Examples

```
## Not run:
## Link in Github repo
download_link(
  link = "https://github.com/fmmattioni/downloadthis/raw/master/inst/example/file_1.pdf",
  button_label = "Download pdf file",
  button_type = "danger",
  has_icon = TRUE,
  icon = "fa fa-save",
  self_contained = FALSE
)

## End(Not run)
```

download_this

Download data frames, lists, or any R object

Description

Wrapper around `bsplus::bs_button()` to provide a download button for HTML outputs in R Markdown. Internally, the function writes the file to `tempdir()`, encodes it, and produces the download button. Currently, Internet Explorer does not support downloading embedded files. For downloading links, files, or directories, see `download_link()`, `download_file()`, and `download_dir()`.

Usage

```
download_this(
  .data,
  ...,
  output_name = NULL,
  output_extension = c(".csv", ".xlsx", ".rds"),
  button_label = "Download data",
  button_type = c("default", "primary", "success", "info", "warning", "danger"),
  icon = "fa fa-save",
  self_contained = FALSE,
  csv2 = TRUE,
  ggsave_args = list()
)
```

Arguments

<code>.data</code>	A data frame or (named) list to write to disk. See 'Examples' for more details.
<code>...</code>	attributes (named arguments) and children (unnamed arguments) of the button, passed to <code>htmltools::tag()</code> .
<code>output_name</code>	Name of the output file, if NULL uses the deparsed <code>.data</code> object.
<code>output_extension</code>	Extension of the output file. Currently, <code>.csv</code> , <code>.xlsx</code> , and <code>.rds</code> are supported. If a (named) list is passed to the function, only <code>.xlsx</code> and <code>.rds</code> are supported.
<code>button_label</code>	Character (HTML), button label
<code>button_type</code>	Character, one of the standard Bootstrap types
<code>icon</code>	Fontawesome tag e.g.: "fa fa-save", set to NULL to
<code>self_contained</code>	A boolean to specify whether your HTML output is self-contained. Default to FALSE.
<code>csv2</code>	A boolean to specify whether to use <code>readr::write_csv2()</code> in case the <code>output_extension</code> is chosen as <code>'csv'</code> . If FALSE, <code>readr::write_csv()</code> will be used instead. Default to TRUE.
<code>ggsave_args</code>	List of arguments to pass to <code>ggplot2::ggsave</code> , e.g.: <code>list(height = 5)</code> .

Value

`htmltools::tag`, `<button>`

Warning

This example will write the `mtcars` dataset to `tempdir()` and produce the download button for the file `mtcars dataset.csv` with the `fa fa-save` icon on the `Download data` label.

Examples

```
## Not run:
# Passing a data frame to the function
mtcars %>%
  download_this(
    output_name = "mtcars dataset",
    output_extension = ".csv",
    button_label = "Download data",
    button_type = "warning",
    has_icon = TRUE,
    icon = "fa fa-save"
  )

# Passing a list with data frames to the function
list(mtcars, iris) %>%
  download_this(
    output_name = "mtcars and iris datasets",
    output_extension = ".xlsx",
    button_label = "Download data",
    button_type = "warning",
    has_icon = TRUE,
    icon = "fa fa-save"
  )

# Passing a named list with data frames to the function
list("mtcars dataset" = mtcars, "iris dataset" = iris) %>%
  download_this(
    output_name = "mtcars and iris datasets",
    output_extension = ".xlsx",
    button_label = "Download data",
    button_type = "warning",
    has_icon = TRUE,
    icon = "fa fa-save"
  )

# Passing any R object to the function
vector_example <- 1:10
linear_model <- lm(mpg ~ gear, data = mtcars)

list(mtcars, iris, vector_example, linear_model) %>%
  download_this(
    output_name = "datasets, vector, and linear model",
    output_extension = ".rds",
    button_label = "Download as rds",
    button_type = "warning",
    has_icon = TRUE,
    icon = "fa fa-save"
  )

## End(Not run)
```

Index

[download_dir](#), [2](#)
[download_file](#), [3](#)
[download_link](#), [4](#)
[download_this](#), [5](#)

[tag](#), [2](#), [3](#), [5](#), [6](#)