

Package ‘htmlreportR’

September 11, 2026

Title 'HTML' Reporting Made Simple(R)

Version 2.0.1

Description

Create compressed, interactive 'HTML' (Hypertext Markup Language) reports with embedded 'Python' code, custom 'JS' ('JavaScript') and 'CSS' (Cascading Style Sheets), and wrappers for 'CanvasXpress' plots, networks and more. Based on <https://pypi.org/project/py-report-html/>, its sister project.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5.0)

Imports methods, mime, ggplot2, knitr, xfun, grDevices, utils,
jsonlite, stringr

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/AEspanMar/htmlreportR>

BugReports <https://github.com/AEspanMar/htmlreportR/issues>

NeedsCompilation no

Author Álvaro Esteban Martos [aut, cph, cre] (ORCID:
<<https://orcid.org/0000-0001-9791-7779>>),
José Córdoba Caballero [aut, cph] (ORCID:
<<https://orcid.org/0000-0002-1821-5742>>),
James Perkins [aut, cph] (ORCID:
<<https://orcid.org/0000-0003-4108-096X>>),
Pedro Seoane Zonjic [aut, cph] (ORCID:
<<https://orcid.org/0000-0002-3020-1415>>),
Jesús Pérez García [aut, cph] (ORCID:
<<https://orcid.org/0009-0006-9388-7465>>)

Maintainer Álvaro Esteban Martos <alvaroesteban@uma.es>

Config/roxygen2/version 8.1.0

Repository CRAN

Date/Publication 2026-09-11 10:20:02 UTC

Contents

col_to_rownames	2
htmlReport-class	2
main_htmlreportR	3
make_head	4
make_html_list	4
parse_paths	5
replace_paired_mark	5
row_to_header	6

Index	7
-------	---

col_to_rownames	<i>col_to_rownames</i>
-----------------	------------------------

Description

turns a column of the input data frame and turns it into its rownames, and then removes the actual column.

Usage

```
col_to_rownames(data_frame, col = 1)
```

Arguments

- data_frame Data frame to manipulate.
- col An integer. Col to set as new rowumns and remove from data frame.

Value

The modified data frame.

Examples

```
col_to_rownames(tail(mtcars))
```

htmlReport-class	<i>htmlReport reference class</i>
------------------	-----------------------------------

Description

An htmlreporter reference class

main_htmlreportR	<i>Build report by loading files from disk and rendering template.</i>
------------------	--

Description

Main htmlreportR script mode function. Renders a template,

Usage

```
main_htmlreportR(options)
```

Arguments

options list containing all necessary elements to build an htmlReport instance.

Details

Main htmlreportR scripting function

Value

None

Examples

```
## Not run:
options <- list(title = "main_htmlreportR example",
  data_files = "path/to/file1", "path/to/fileN",
  output_file = "path/to/output",
  source_folder = "path/to/package/scripts",
  compress_obj = TRUE,
  css_files = "path/to/custom/css",
  js_files = "path/to/custom/js",
  cdn_css = "path/to/cdn_css",
  cdn_js = "path/to/cdn_js",
  menu = "menu")
main_htmlreportR (options)

## End(Not run)
```

make_head	<i>Make HTML Report Head</i>
-----------	------------------------------

Description

This method generates the head section of an HTML report by adding the title to an `htmlReport` object.

Details

Make HTML Report Head

Value

An updated `htmlReport` object with the title added to its head section.

make_html_list	<i>make_html_list</i>
----------------	-----------------------

Description

Take an input vector or list and use it to build an html list of specified type

Usage

```
make_html_list(
  list_content,
  list_levels = NULL,
  list_types = NULL,
  default_type = "ul"
)
```

Arguments

<code>list_content</code>	Vector or list of elements with which to build the list
<code>list_levels</code>	Vector or list defining nesting levels. Default NULL
<code>list_types</code>	Vector or list with types to assign to each element ("ul" for unordered or "ol" for ordered). If not supplied, <code>default_type</code> will determine list type for each element. Default NULL
<code>default_type</code>	List type to default if <code>list_types</code> is undefined, "ul" for unordered or "ol" for ordered. Default "ul"

Value

Formatted html list ready to render.

Examples

```
content <- c("One", "Two", "Three")
make_html_list(list_content = content)
```

parse_paths	<i>parse_paths</i>
-------------	--------------------

Description

takes a vector of comma-separated paths and parses it so files can be loaded. If any of the paths contain wildcards, they will be expanded.

Usage

```
parse_paths(string)
```

Arguments

string	String containing paths to expand.
--------	------------------------------------

Value

Every path found with provided expression.

Examples

```
dir <- file.path(find.package("htmlreportR"), "tests/*")
parse_paths(dir)
```

replace_paired_mark	<i>replace_paired_mark</i>
---------------------	----------------------------

Description

Recursively replaces paired marks in a string.

Usage

```
replace_paired_mark(string, pattern, replace)
```

Arguments

string	String to edit
pattern	Pattern to recursively replace
replace	Vector containing, in that order, expression that will replace the first paired element and the second paired element

Value

Modified string.

Examples

```
example_string <- "** This should be in bold **"
pattern <- "(\\*\\*+?)([- \\w]+)(\\*\\*+?)"
replace_paired_mark(example_string, pattern, c("<strong>", "</strong>"))
```

row_to_header	<i>row_to_header</i>
---------------	----------------------

Description

turns a row of the input data frame and turns it into its colnames, and then removes the actual row.

Usage

```
row_to_header(data_frame, row = 1)
```

Arguments

- data_frame Data frame to manipulate.
- row An integer. Row to set as new columns and remove from data frame.

Value

The modified data frame.

Examples

```
row_to_header(head(mtcars))
```

Index

`col_to_rownames`, [2](#)

`htmlReport (htmlReport-class)`, [2](#)

`htmlReport-class`, [2](#)

`main_htmlreportR`, [3](#)

`make_head`, [4](#)

`make_html_list`, [4](#)

`parse_paths`, [5](#)

`replace_paired_mark`, [5](#)

`row_to_header`, [6](#)