Package 'nestable'

May 9, 2026

Type Package

Title Collapsible 'HTML' Tables from Hierarchical Data

Version 0.1.0

Description Creates collapsible, expandable 'HTML' tables from hierarchical data. Supports data frame input with multi-level grouping, custom column formatters, bottom-up rollup aggregation, and CSS-variable-based theming. Works in 'Shiny' applications, R Markdown, 'Quarto', and the 'RStudio' Viewer.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports htmltools (>= 0.5.0)

Suggests shiny (>= 1.7.0), testthat (>= 3.0.0)

NeedsCompilation no

Author Derek Underwood [aut, cre]

Maintainer Derek Underwood <dereku@gmail.com>

Repository CRAN

Date/Publication 2026-04-28 19:00:10 UTC

Contents

col_def 2
df_to_tree 3
fmt_currency 4
fmt_percent 4
nestable 5
nestableOutput 6
nestable_theme 6
node 8
rows_to_nodes 8
weighted_rollup 9

Index **10**

`col_def` *Define a table column*

Description

Define a table column

Usage

```
col_def(
  key,
  header = NULL,
  format = function(x) base::format(x),
  color = NULL,
  rollup = "sum",
  width = NULL
)
```

Arguments

`key`	Character. The value key — the column name in your data frame (or the name used in each node's `.values` list).
`header`	Character. Column header text. Defaults to a title-cased version of `key` (e.g. `"market_cap"` → `"Market Cap"`).
`format`	Function `function(x) -> character` for display formatting. Defaults to `base::format()`.
`color`	Function `function(x) -> CSS color string`, or `NULL`. Default `NULL`.
`rollup`	How parent rows are aggregated. Either a shortcut string (`"sum"` or `"mean"`) or a function `function(vals, child_values) -> scalar`. `vals` is a numeric vector of children's values for this column; `child_values` is the full list of each child's value lists (useful for weighted aggregation via `weighted_rollup()`). Defaults to `"sum"`.
`width`	CSS width string (e.g. `"120px"`, `"10%"`) applied to the column header and every data cell. `NULL` (default) leaves width unset.

Value

A named list describing the column.

`df_to_tree` *Convert a flat data frame into a nested node tree*

Description

Convert a flat data frame into a nested node tree

Usage

```
df_to_tree(
  df,
  name_col,
  value_cols,
  group_col = NULL,
  total = NULL,
  node_values = list()
)
```

Arguments

`df`	A data frame.
`name_col`	Column name to use as the node label (leaf rows).
`value_cols`	Character vector of value column names.
`group_col`	Character vector of grouping columns, outermost first. Each element adds one nesting level. `NULL` returns a flat list of leaves.
`total`	Optional string. When non-`NULL` a single root node with this label wraps the entire tree (grand-total row). `NULL` for no total.

`node_values`	Optional named list of pre-supplied values for group (and total) nodes. Each name is a node label; each value is a named list of column values that should be displayed as-is rather than rolled up from children. Useful when aggregated figures (e.g. time-weighted returns) are already known and differ from a simple weighted average of the leaves.
	Example — supply a pre-computed return for the "Technology" sector and the "Mag 7" grand total:
	`node_values = list(`
	`  "Technology" = list(ytd_return = 2.5),`
	`  "Mag 7"      = list(ytd_return = 4.1)`
	`)`
	Any column *not* listed for a node still falls back to rollup from children.

Value

A list of `node()` objects suitable for passing to `nestable()`.

fmt_currency *Currency format function factory*

Description

Returns a formatting function for use as `format_fn` in `col_def()`.

Usage

```
fmt_currency(prefix = "$", suffix = "", digits = 2L, big_mark = ",")
```

Arguments

`prefix`	Character prepended before the number. Default `"$"`.
`suffix`	Character appended after the number. Default `""`.
`digits`	Integer decimal places. Default `2L`.
`big_mark`	Thousands separator. Default `","`.

Value

A function `function(x) -> character`.

fmt_percent *Percentage format function factory*

Description

Returns a formatting function for use as `format_fn` in `col_def()`.

Usage

```
fmt_percent(digits = 2L, plus = TRUE)
```

Arguments

`digits`	Integer decimal places. Default `2L`.
`plus`	Logical. Prefix non-negative values with `"+"`. Default `TRUE`.

Value

A function `function(x) -> character`.

nestable *Create a nestable collapsible HTML table*

Description

Create a nestable collapsible HTML table

Usage

```
nestable(
  data_root,
  columns,
  theme = nestable_theme(),
  name_col = "name",
  name_header = NULL,
  name_col_width = NULL,
  uid = new_widget_uid()
)
```

Arguments

data_root	A list of top-level node() objects. Build with node(), rows_to_nodes(), or df_to_tree().
columns	Column specification. Three forms are accepted: • A character vector of key names: c("market_cap", "ytd_return") — headers are auto-derived from the key (e.g. "market_cap" → "Market Cap"). • A *named* character vector: c("Market Cap" = "market_cap", "YTD Return" = "ytd_return") — explicit headers, default formatting and rollup. • A list of col_def() objects for full control over formatting, colours, and rollup behaviour.
theme	A theme list from nestable_theme().
name_col	Character. The node label key — the name_col used when building the tree with df_to_tree(), or "name" when constructing nodes manually. Used to auto-derive name_header via title-casing when name_header is NULL. Default "name".
name_header	Character. Header label for the first (name/label) column. NULL (default) derives the label from name_col (e.g. "security_name" → "Security Name").
name_col_width	CSS width string (e.g. "200px", "30%") applied to the name column header and every name cell. NULL (default) leaves the width unset, allowing the browser to size the column automatically.
uid	Character. Widget UID prefix for HTML element id attributes. Defaults to a random string so multiple tables on one page never clash. Override only when reproducible IDs are needed (e.g. tests).

Value

An `htmltools::browsable()` `tagList`. Renders inline in R Markdown, Quarto, and the RStudio Viewer; use inside `shiny::renderUI()` or `renderNestable()` in Shiny apps.

---

nestableOutput *Shiny UI output for a nestable table*

Description

Use with `renderNestable()` in the server. These are thin wrappers over `shiny::uiOutput()` and `shiny::renderUI()` — no `htmlwidgets` dependency is required.

Usage

```
nestableOutput(outputId, ...)

renderNestable(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	The output variable name.
`...`	Additional arguments passed to `shiny::uiOutput()`.
`expr`	An expression returning a `nestable()` widget.
`env`	The environment in which to evaluate `expr`.
`quoted`	Logical. Is `expr` already quoted? Default `FALSE`.

Value

A Shiny UI element.

A Shiny render function.

---

nestable_theme *Create a nestable theme*

Description

Every argument maps to a CSS custom property (`--ntbl-*`) set inline on the widget's wrapper `<div>`, so multiple instances with different themes can coexist on the same page without conflict.

Usage

```
nestable_theme(
  title = "",
 font_family = "-apple-system, BlinkMacSystemFont, \"Segoe UI\", Roboto, sans-serif",
  font_size = "14px",
  table_bg = "#ffffff",
  table_shadow = "0 1px 4px rgba(0,0,0,.12)",
  table_radius = "6px",
  table_max_w = "680px",
  header_bg = "#37474f",
  header_color = "#ffffff",
  row_border = "#eceff1",
  row_hover_bg = "#f9fbe7",
  parent_weight = "600",
  toggle_color = "#546e7a",
  indent_px = 20L,
  zoom = "normal"
)
```

Arguments

`title`	Character. Title shown above the table. Default `""` (no title).
`font_family`	CSS font-family string.
`font_size`	CSS font-size string. Default `"14px"`.
`table_bg`	Table background colour. Default `"#ffffff"`.
`table_shadow`	CSS box-shadow for the table.
`table_radius`	CSS border-radius. Default `"6px"`.
`table_max_w`	CSS max-width. Default `"680px"`.
`header_bg`	Header row background. Default `"#37474f"`.
`header_color`	Header row text colour. Default `"#ffffff"`.
`row_border`	Row separator colour. Default `"#eceff1"`.
`row_hover_bg`	Row hover background. Default `"#f9fbe7"`.
`parent_weight`	CSS font-weight for parent rows. Default `"600"`.
`toggle_color`	Colour of the expand/collapse arrow. Default `"#546e7a"`.
`indent_px`	Integer pixels of indentation per nesting level. Default `20L`.
`zoom`	CSS zoom level applied to the entire widget. Accepts any valid CSS `zoom` value: a number (`1.25`), a percentage (`"125%"`), or `"normal"` (default). Useful for global size/scale adjustments without touching individual font-size or dimension settings.

Value

A named list of theme values.

node *Define a tree node*

Description

Define a tree node

Usage

```
node(name, ..., .values = list())
```

Arguments

`name`	Display label shown in the Name column.
`...`	Child `node()` objects. Supplying children makes this a parent (group) row whose column values are rolled up from children unless overridden via `.values`.
`.values`	Named list of column values. For leaf nodes supply all values here. For parent nodes any value supplied here overrides the computed rollup for that column; omitted columns are still computed from children.

Value

A named list with elements `name`, `values`, and `children`.

rows_to_nodes *Convert data frame rows into leaf nodes*

Description

Convert data frame rows into leaf nodes

Usage

```
rows_to_nodes(df, name_col, value_cols)
```

Arguments

`df`	A data frame.
`name_col`	Column name to use as the node label.
`value_cols`	Character vector of column names to carry as `.values`.

Value

A list of `node()` objects.

`weighted_rollup` *Weighted-average rollup function factory*

Description

Returns a rollup function for use as `rollup_fn` in `col_def()`. Computes the weighted average of `vals` using another key's values as weights.

Usage

```
weighted_rollup(weight_key)
```

Arguments

`weight_key`	Character. The value key to use as weights (e.g. `"market_cap"`). Each child's value for this key is used as its weight.

Value

A function `function(vals, child_values) -> numeric`.

Index

base::format(), *2*

col_def, 2
col_def(), *4, 5*, *9*

df_to_tree, 3
df_to_tree(), *5*

fmt_currency, 4
fmt_percent, 4

htmltools::browsable(), *6*

nestable, 5
nestable(), *3*, *6*
nestable_theme, 6
nestable_theme(), *5*
nestableOutput, 6
node, 8
node(), *3*, *5*, *8*

renderNestable *(*nestableOutput*)*, 6
renderNestable(), *6*
rows_to_nodes, 8
rows_to_nodes(), *5*

shiny::renderUI(), *6*
shiny::uiOutput(), *6*

weighted_rollup, 9
weighted_rollup(), *2*