Package ‘TT’

October 12, 2022

Title Display Tree Structured Data using Datatable Widget (DT)

Version 0.98

Description Wrapper of datatable widget, allowing display of data.tree objects.
   All arguments of the data.tree become columns and each node is a row.
   Adds column with buttons allowing folding and unfolding the levels.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 2.10)

Imports DT, dplyr (>= 1.0.0), magrittr, purrr, data.tree, htmlwidgets

NeedsCompilation no

Author Michal Zielaskowski [aut, cre]

Maintainer Michal Zielaskowski <michal.zielaskowski@gmail.com>

Repository CRAN

Date/Publication 2022-04-05 20:42:29 UTC

R topics documented:

   col_order ................................................................. 2
   formatCurrency .......................................................... 2
   org ................................................................. 4
   treetable ............................................................... 5

Index 7
**Description**

class vector to demonstrate ordering option. See example of `treetable`

**Usage**

col_order

**Format**

class vector

---

**formatCurrency**

**Format table columns**

**Description**

Simply wrapper on format... family functions of ’DT’ package. For details see: `formatCurrency`
The wrappers are not affecting behavior of original format... functions

**Usage**

```r
formatCurrency(
  table,
  columns,
  currency = "$",
  interval = 3,
  mark = ",",
  digits = 2,
  dec.mark = getOption("OutDec"),
  before = TRUE
)

formatDate(table, columns, method = "toDateString", params = NULL)

formatPercentage(
  table,
  columns,
  digits = 0,
  interval = 3,
  mark = ",",
  dec.mark = getOption("OutDec")
)```

```r```
formatCurrency

formatRound(
    table,
    columns,
    digits = 2,
    interval = 3,
    mark = ",",
    dec.mark = getOption("OutDec")
)

formatSignif(
    table,
    columns,
    digits = 2,
    interval = 3,
    mark = ",",
    dec.mark = getOption("OutDec")
)

formatString(table, columns, prefix = "", suffix = "")

formatStyle(
    table,
    columns,
    valueColumns = columns,
    target = c("cell", "row"),
    fontWeight = NULL,
    color = NULL,
    backgroundColor = NULL,
    background = NULL,
    ...
)

Arguments

    table         a table object created from datatable()
    columns       the indices of the columns to be formatted (can be character, numeric, logical,
                   or a formula of the form ~ V1 + V2, which is equivalent to c('V1', 'V2'))
    currency      the currency symbol
    interval      put a marker after how many digits of the numbers
    mark          the marker after every interval decimals in the numbers
    digits        the number of decimal places to round to
    dec.mark      a character to indicate the decimal point
    before        whether to place the currency symbol before or after the values
method: The method(s) to convert a date to string in JavaScript; see DT:::DateMethods for a list of possible methods, and https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date for a full reference.

params: A list of parameters for the specific date conversion method, e.g., for the toLocaleDateString() method, your browser may support params = list('ko-KR', list(year = 'numeric', month = 'long', day = 'numeric'))

prefix: String to put in front of the column values

suffix: String to put after the column values

valueColumns: Indices of the columns from which the cell values are obtained; this can be different with the columns argument, e.g., you may style one column based on the values of a different column.

target: The target to apply the CSS styles to (the current cell or the full row)

fontWeight: The font weight, e.g., 'bold' and 'normal'

color: The font color, e.g., 'red' and '#ee00aa'

backgroundColor: The background color of table cells

background: The background of table cells

...: Other CSS properties, e.g., 'border', 'font-size', 'text-align', and so on; if you want to condition CSS styles on the cell values, you may use the helper functions such as styleInterval(); note the actual CSS property names are dash-separated, but you can use camelCase names in this function (otherwise you will have to use backticks to quote the names, e.g., 'font-size' = '12px'), and this function will automatically convert camelCase names to dash-separated names (e.g., 'fontWeight' will be converted to 'font-weight' internally)

Value

Return formatted 'HTML' widget of 'DataTables'

Example data for TT package

Description

'data.tree' object that can be used as an example to see how table widget looks like. Each node of 'data.tree' store attributes (dates) with some numbers.

Usage

org
**Format**

'\texttt{data.tree}' object with 7 attributes in each node, value of the leaf attributes is some random number, parents value is cumulative sum of children The attributes are:

- name of the node
- six consecutive months of the year (Jan-2021, Feb-2021...)

**Description**

Wrapper of 'datatable' widget, allowing display of 'data.tree' objects. All arguments of the 'data.tree' become columns and each node is a row. Adds column with buttons allowing folding and unfolding the levels.

**Usage**

\begin{verbatim}
treetable(data, color = "#0177A5", colnames = list(), ...)
\end{verbatim}

**Arguments**

- **data** \texttt{data.tree} object. \texttt{treetable} will extract all arguments in alphabetical order - these will be a columns. For renaming and ordering of the columns see colnames.
- **color** base color (hue) to color the table. Each level will differ with saturation and luminosity.
- **colnames** if list() of characters provided, arguments of data.tree (columns) will be renamed. If vector() provided, columns will be renamed as for list input, additionally columns will be reordered according to vector level after renaming.
- ... \texttt{datatable} parameters

**Details**

Package consist of treetable function (wrapper of 'datatable') that convert data.tree object to 'dataframe' and 'JS' callback function called after creating the table. Treetable function ads hidden columns used by 'JS' for formatting and folding/unfolding level rows. Hidden columns shall be completely transparent for user.

Package also include 'DT::format...' functions wrappers, which are working exactly as originals, but are necessary to protect special (helper) columns used by 'JS' callback function for formatting.

Color formatting is done by 'kolorWheel' 'JS' script done by Zalka Erno
e-mail: ern0[at]linkbroker.hu
http://linkbroker.hu/stuff/kolorwheel.js/
Value

Return 'HTML' widget using the 'JavaScript' library 'DataTables'

References

https://github.com/zielaskowski/tree-table

See Also

datatable
data.tree

Examples

data("org")
data("col_order")
colnames <- factor(c("org",org$attributesAll),
                   levels = col_order)
treetable(org, color='#FFFFFF', colnames=colnames)

# still datatable works as expected when data.frame provided
treetable(data.frame(  
    date = seq(as.Date("2015-01-01"), by = "day", length.out = 5), x = 1:5))
Index

* datasets
  col_order, 2
  org, 4
  col_order, 2

data.tree, 6
datatable, 3, 5, 6

formatCurrency, 2, 2
formatDate (formatCurrency), 2
formatPercentage (formatCurrency), 2
formatRound (formatCurrency), 2
formatSignif (formatCurrency), 2
formatString (formatCurrency), 2
formatStyle (formatCurrency), 2

org, 4

styleInterval, 4

treetable, 2, 5