Package 'cheetahR'

May 12, 2025

Title High Performance Tables Using 'Cheetah Grid'

Version 0.2.0

License GPL (>= 3)

Description An R interface to 'Cheetah Grid', a high-performance JavaScript table widget. 'cheetahR' allows users to render millions of rows in just a few milliseconds, making it an excellent alternative to other R table widgets. The package wraps the 'Cheetah Grid' JavaScript functions and makes them readily available for R users. The underlying grid implementation is based on 'Cheetah Grid' <https://github.com/future-architect/cheetah-grid>.

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports htmlwidgets, jsonlite, tibble

Suggests testthat (>= 3.0.0), utils, shiny, quarto, knitr, dplyr, palmerpenguins

Config/testthat/edition 3

VignetteBuilder quarto

NeedsCompilation no

Author Olajoke Oladipo [aut, cre], David Granjon [aut], cynkra GmbH [fnd]

Maintainer Olajoke Oladipo <olajoke@cynkra.com>

Repository CRAN

Date/Publication 2025-05-12 15:40:08 UTC

Contents

`add_cell_message` *Create a JavaScript cell message function for cheetahR widgets*

Description

Generates a JS function (wrapped with `htmlwidgets::JS`) that, given a record (`rec`), returns an object with `type` and `message`.

Usage

```
add_cell_message(type = c("error", "warning", "info"), message = "message")
```

Arguments

`type`	A string that specifies the type of message. One of `"error"`, `"warning"`, or `"info"`. Default is `"error"`.
`message`	A string or JS expression. If it contains `rec.`, `?`, `:`, or a trailing `;`, it is treated as raw JS (no additional quoting). Otherwise, it is escaped and wrapped in single quotes.

Value

A `htmlwidgets::JS` object containing a JavaScript function definition:

```
function(rec) {
  return {
    type: "<type>",
    message: <message>
  };
}
```

Use this within `column_def()` for cell validation

Examples

```
set.seed(123)
iris_rows <- sample(nrow(iris), 10)
data <- iris[iris_rows, ]

# Simple warning
cheetah(
  data,
  columns = list(
    Species = column_def(
```

```
      message = add_cell_message(type = "info", message = "Ok")
    )
  )
)

# Conditional error using `js_ifelse()`
cheetah(
  data,
  columns = list(
    Species = column_def(
      message = add_cell_message(
        message = js_ifelse(Species == "setosa", "", "Invalid")
      )
    )
  )
)

# Directly using a JS expression as string
cheetah(
  data,
  columns = list(
    Sepal.Width = column_def(
      style = list(textAlign = "left"),
      message = add_cell_message(
        type = "warning",
        message = "rec['Sepal.Width'] <= 3 ? 'NarrowSepal' : 'WideSepal';"
      )
    )
  )
)
```

---

cheetah *Create a Cheetah Grid widget*

Description

Creates a high-performance table widget using Cheetah Grid.

Usage

```
cheetah(
  data,
  columns = NULL,
  column_group = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL,
  rownames = TRUE,
```

```
  search = c("disabled", "exact", "contains"),
  sortable = TRUE
)
```

Arguments

`data`	A data frame or matrix to display
`columns`	A list of column definitions. Each column can be customized using `column_def()`.
`column_group`	A list of column groups. Each group can be customized using
`width`	Width of the widget
`height`	Height of the widget
`elementId`	The element ID for the widget
`rownames`	Logical. Whether to show rownames. Defaults to TRUE.
`search`	Whether to enable a search field on top of the table. Default to `disabled`. Use `exact` for exact matching or `contains` to get larger matches.
`sortable`	Logical. Whether to enable sorting on all columns. Defaults to TRUE.

Value

An HTML widget object of class 'cheetah' that can be:

- Rendered in R Markdown documents
- Used in Shiny applications
- Displayed in R interactive sessions

The widget renders as an HTML table with all specified customizations.

`cheetah-shiny` *Shiny bindings for cheetah*

Description

Output and render functions for using cheetah within Shiny applications and interactive Rmd documents.

Usage

```
cheetahOutput(outputId, width = "100%", height = "400px")

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

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a cheetah
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

Value

cheetahOutput returns a Shiny output function that can be used in the UI definition. renderCheetah returns a Shiny render function that can be used in the server definition.

---

column_def *Column definition utility*

Description

Needed by cheetah to customize columns individually.

Usage

```
column_def(
  name = NULL,
  width = NULL,
  min_width = NULL,
  max_width = NULL,
  column_type = NULL,
  action = NULL,
  menu_options = NULL,
  style = NULL,
  message = NULL,
  sort = FALSE
)
```

Arguments

name	Custom name.
width	Column width.
min_width	Column minimal width.
max_width	Column max width.
column_type	Column type. By default, the column type is inferred from the data type of the column. There are 7 possible options:

- "text" for text columns.
- "number" for numeric columns.
- "check" for check columns.
- "image" for image columns.
- "radio" for radio columns.
- "multilinetext" for multiline text columns.
- "menu" for menu selection columns. If column_type == "menu", action parameter must be set to "inline_menu" and menu_options must be provided. Note: Works efficiently only in shiny.

action — The action property defines column actions. Select the appropriate Action class for the column type.

- "input" for input action columns.
- "check" for check action columns.
- "radio" for radio action columns.
- "inline_menu" for menu selection columns.

menu_options — A list of menu options when using column_type = "menu". Each option should be a list with value and label elements. The menu options must be a list of lists, each containing a value and label element. The label element is the label that will be displayed in the menu.

style — Column style.

message — Cell message. Expect a htmlwidgets::JS() function that takes rec as argument. It must return an object with two properties: type for the message type ("info", "warning", "error") and the message that holds the text to display. The latter can leverage a JavaScript ternary operator involving rec.<COLNAME> (COLNAME being the name of the column for which we define the message) to check whether the predicate function is TRUE. You can also use add_cell_message() to generated the expected JS expression. See details for example of usage.

sort — Whether to sort the column. Default to FALSE. May also be a JS callback to create custom logic (does not work yet).

Details

Cell messages:

When you write a message, you can pass a function like so:

```
<COLNAME> = column_def(
  action = "input",
   message = JS(
    "function(rec) {
       return {
         //info message
         type: 'info',
         message: rec.<COLNAME> ? null : 'Please check.',
       }
     }")
  )
```

Or use add_cell_message():

```
<COLNAME> = column_def(
  action = "input",
   message = add_cell_message(type = "info", message = "Ok")
  )
```

See add_cell_message() for more details.

Value

A list of column options to pass to the JavaScript API.

Examples

```
cheetah(
  iris,
  columns = list(
    Sepal.Length = column_def(name = "Length"),
    Sepal.Width = column_def(name = "Width"),
    Petal.Length = column_def(name = "Length"),
    Petal.Width = column_def(name = "Width")
  )
)
```

---

column_group *Column group definitions*

Description

Creates a column group definition for grouping columns in a Cheetah Grid widget.

Usage

```
column_group(name = NULL, columns, header_style = NULL)
```

Arguments

name	Character string. The name to display for the column group.
columns	Character vector. The names of the columns to include in this group.
header_style	Named list of possibleCSS style properties to apply to the column group header.

Value

A list containing the column group definition.

Examples

```
cheetah(
  iris,
  columns = list(
    Sepal.Length = column_def(name = "Length"),
    Sepal.Width = column_def(name = "Width"),
    Petal.Length = column_def(name = "Length"),
    Petal.Width = column_def(name = "Width")
  ),
  column_group = list(
    column_group(name = "Sepal", columns = c("Sepal.Length", "Sepal.Width")),
    column_group(name = "Petal", columns = c("Petal.Length", "Petal.Width"))
  )
)
```

---

js_ifelse *Convert an R logical expression into a JS ternary expression*

Description

Convert an R logical expression into a JS ternary expression

Usage

```
js_ifelse(condition, if_true = "", if_false = "")
```

Arguments

condition	An R logical expression (supports %in% / %notin% / grepl() / comparisons / & |)
if_true	String to return when the condition is TRUE. Default is an empty string, which interprets as null in JS.
if_false	String to return when the condition is FALSE. Default is an empty string, which interprets as null in JS.

Value

A single character string containing a JavaScript ternary expression.

Index