

Package ‘rhandsontable’

August 25, 2015

Title Interface to the 'Handsontable.js' Library

Version 0.2

Maintainer Jonathan Owen <jonathanro@gmail.com>

Description An R interface to the 'Handsontable' JavaScript library, which is a minimalist Excel-like data grid editor.

License MIT + file LICENSE

URL <http://jrowen.github.io/rhandsontable/>

BugReports <https://github.com/jrowen/rhandsontable/issues>

Imports jsonlite, htmlwidgets (>= 0.3.3), magrittr, methods

Suggests knitr

Enhances shiny (>= 0.12)

VignetteBuilder knitr

NeedsCompilation no

Author Jonathan Owen [aut, cre, cph],
Jeff Allen [ctb],
Yihui Xie [ctb],
Enzo Martoglio [ctb],
Warpechowski Marcin [ctb, cph] (Handsontable.js library),
Handsoncode sp. z o.o. [ctb, cph] (Handsontable.js library),
Aisch Gregor [ctb, cph] (Chroma.js library),
Wood Tim [ctb, cph] (Moment.js library),
Chernev Iskren [ctb, cph] (Moment.js library),
Moment.js contributors [ctb, cph] (Moment.js library),
Bushell David [ctb, cph] (Pikaday.js library),
jQuery Foundation [ctb, cph] (jQuery.js library),
Splunk Inc [ctb, cph] (Sparkline.js library)

Repository CRAN

Date/Publication 2015-08-25 08:54:03

R topics documented:

| | |
|----------------------------------|----|
| rhandsontable-package | 2 |
| hot_cell | 2 |
| hot_col | 3 |
| hot_cols | 4 |
| hot_heatmap | 5 |
| hot_rows | 6 |
| hot_table | 6 |
| hot_to_r | 8 |
| hot_validate_character | 8 |
| hot_validate_numeric | 9 |
| renderRHandsontable | 10 |
| rhandsontable | 10 |
| rhandsontable-exports | 11 |
| rHandsontableOutput | 12 |

| | |
|--------------|-----------|
| Index | 13 |
|--------------|-----------|

rhandsontable-package *rhandsontable*

Description

R interface for creating tables using Handsontable, [urlhttp://http://handsontable.com/](http://http://handsontable.com/)

Details

For full documentation on the package, visit <http://jrowen.github.io/rhandsontable/>

hot_cell *Handsontable widget*

Description

Configure single cell. See [Handsontable.js](#) for details.

Usage

```
hot_cell(hot, row, col, comment = NULL)
```

Arguments

| | |
|---------|----------------------------------|
| hot | rhandsontable object |
| row | numeric row index |
| col | numeric column index |
| comment | character comment to add to cell |

See Also[hot_cols](#), [hot_rows](#)**Examples**

```
library(rhandsontable)
DF = data.frame(val = 1:10, bool = TRUE, big = LETTERS[1:10],
               small = letters[1:10],
               dt = seq(from = Sys.Date(), by = "days", length.out = 10),
               stringsAsFactors = FALSE)

rhandsontable(DF, readOnly = TRUE) %>%
  hot_cell(1, 1, "Test comment")
```

| | |
|---------|----------------------------|
| hot_col | <i>Handsontable widget</i> |
|---------|----------------------------|

Description

Configure single column.

Usage

```
hot_col(hot, col, type = NULL, format = NULL, source = NULL,
        strict = NULL, readOnly = NULL, validator = NULL, allowInvalid = NULL,
        halign = NULL, valign = NULL, renderer = NULL)
```

Arguments

| | |
|--------------|--|
| hot | rhandsontable object |
| col | vector of column names or indices |
| type | character specify the data type. Options include: numeric, date, checkbox, select, dropdown, autocomplete, password, and handsontable (not implemented yet) |
| format | characer specifying column format. See Cell Types at Handsontable.js for the formatting options for each data type. Numeric columns are formatted using Numeral.js . |
| source | a vector of choices for select, dropdown and autocomplete column types |
| strict | logical specifying whether values not in the source vector will be accepted |
| readOnly | logical making the table read-only |
| validator | character defining a Javascript function to be used to validate user input. See hot_validate_numeric and hot_validate_character for pre-build validators. |
| allowInvalid | logical specifying whether invalid data will be accepted. Invalid data cells will be color red. |

| | |
|----------|--|
| halign | character defining the horizontal alignment. Possible values are htLeft, htCenter, htRight and htJustify |
| valign | character defining the vertical alignment. Possible values are htTop, htMiddle, htBottom |
| renderer | character defining a Javascript function to be used to format column cells. Can be used to implement conditional formatting. |

See Also

[hot_cols](#), [hot_rows](#), [hot_cell](#)

Examples

```
library(rhandsontable)
DF = data.frame(val = 1:10, bool = TRUE, big = LETTERS[1:10],
               small = letters[1:10],
               dt = seq(from = Sys.Date(), by = "days", length.out = 10),
               stringsAsFactors = FALSE)

rhandsontable(DF, rowHeaders = NULL) %>%
  hot_col(col = "big", type = "dropdown", source = LETTERS) %>%
  hot_col(col = "small", type = "autocomplete", source = letters,
         strict = FALSE)
```

| | |
|----------|----------------------------|
| hot_cols | <i>Handsontable widget</i> |
|----------|----------------------------|

Description

Configure multiple columns.

Usage

```
hot_cols(hot, colWidths = NULL, columnSorting = NULL,
         manualColumnMove = NULL, manualColumnResize = NULL,
         fixedColumnsLeft = NULL, ...)
```

Arguments

| | |
|--------------------|--|
| hot | rhandsontable object |
| colWidths | a scalar or numeric vector of column widths |
| columnSorting | logical enabling row sorting. Sorting only alters the table presentation and the original dataset row order is maintained. |
| manualColumnMove | logical enabling column drag-and-drop reordering |
| manualColumnResize | logical enableline column width resizing |

```

fixedColumnsLeft
    a numeric vector indicating which columns should be frozen on the left
...
    passed to hot_col

```

See Also

[hot_col](#), [hot_rows](#), [hot_cell](#)

Examples

```

library(rhandsontable)
DF = data.frame(val = 1:10, bool = TRUE, big = LETTERS[1:10],
               small = letters[1:10],
               dt = seq(from = Sys.Date(), by = "days", length.out = 10),
               stringsAsFactors = FALSE)

rhandsontable(DF) %>%
  hot_cols(columnSorting = TRUE)

```

| | |
|-------------|----------------------------|
| hot_heatmap | <i>Handsontable widget</i> |
|-------------|----------------------------|

Description

Add heatmap to table. See [Heatmaps for values in a column](#) for details.

Usage

```
hot_heatmap(hot, cols, color_scale = c("#ED6D47", "#17F556"),
            renderer = NULL)
```

Arguments

| | |
|-------------|--|
| hot | rhandsontable object |
| cols | numeric vector of columns to include in the heatmap. If missing all columns are used. |
| color_scale | character vector that includes the lower and upper colors |
| renderer | character defining a Javascript function to be used to determine the cell colors. If missing, <code>rhandsontable:::renderer_heatmap</code> is used. |

Examples

```

MAT = matrix(rnorm(50), nrow = 10, dimnames = list(LETTERS[1:10],
                                                  letters[1:5]))

rhandsontable(MAT) %>%
  hot_heatmap()

```

| | |
|----------|----------------------------|
| hot_rows | <i>Handsontable widget</i> |
|----------|----------------------------|

Description

Configure rows. See [Handsontable.js](#) for details.

Usage

```
hot_rows(hot, rowHeights = NULL, fixedRowsTop = NULL)
```

Arguments

| | |
|--------------|--|
| hot | rhandsontable object |
| rowHeights | a scalar or numeric vector of row heights |
| fixedRowsTop | a numeric vector indicating which rows should be frozen on the top |

See Also

[hot_cols](#), [hot_cell](#)

Examples

```
library(rhandsontable)
MAT = matrix(rnorm(50), nrow = 10, dimnames = list(LETTERS[1:10],
  letters[1:5]))

rhandsontable(MAT, width = 300, height = 150) %>%
hot_cols(colWidths = 100, fixedColumnsLeft = 1) %>%
hot_rows(rowHeights = 50, fixedRowsTop = 1)
```

| | |
|-----------|----------------------------|
| hot_table | <i>Handsontable widget</i> |
|-----------|----------------------------|

Description

Configure table. See [Handsontable.js](#) for details.

Usage

```
hot_table(hot, contextMenu = TRUE, stretchH = "none", allowRowEdit = TRUE,
  allowColEdit = TRUE, resizeOnRowEdit = TRUE, resizeOnColEdit = TRUE,
  customBorders = NULL, groups = NULL, highlightRow = NULL,
  highlightCol = NULL, comments = NULL, exportToCsv = NULL,
  csvFileName = "download.csv", ...)
```

Arguments

| | |
|-----------------|---|
| hot | rhandsontable object |
| contextMenu | logical enabling the right-click menu |
| stretchH | character describing column stretching. Options are 'all', 'right', and 'none'. See Column stretching for details. |
| allowRowEdit | logical enabling right-click row options |
| allowColEdit | logical enabling right-click column options |
| resizeOnRowEdit | logical resize table when add/removing rows |
| resizeOnColEdit | logical resize table when add/removing rows |
| customBorders | json object. See Custom borders for details. |
| groups | json object. See Grouping & ungrouping of rows and columns for details. |
| highlightRow | logical enabling row highlighting for the selected cell |
| highlightCol | logical enabling column highlighting for the selected cell |
| comments | logical enabling comments in the table, including the corresponding options in the right-click menu. User comments are not currently returned to R. |
| exportToCsv | logical adding a context menu option to export the table data to a csv file |
| csvFileName | character csv file name |
| ... | passed to Handsontable.js constructor |

See Also

[rhandsontable](#)

Examples

```
library(rhandsontable)
DF = data.frame(val = 1:10, bool = TRUE, big = LETTERS[1:10],
               small = letters[1:10],
               dt = seq(from = Sys.Date(), by = "days", length.out = 10),
               stringsAsFactors = FALSE)

rhandsontable(DF) %>%
hot_table(highlightCol = TRUE, highlightRow = TRUE,
          allowRowEdit = FALSE, allowColEdit = FALSE)
```

| | |
|----------|----------------------------|
| hot_to_r | <i>Handsontable widget</i> |
|----------|----------------------------|

Description

Convert handsontable data to R object. Can be used in a shiny app to convert the input json to an R dataset.

Usage

```
hot_to_r(...)
```

Arguments

... passed to `rhandsontable:::toR`

See Also

[rHandsontableOutput](#)

| | |
|------------------------|----------------------------|
| hot_validate_character | <i>Handsontable widget</i> |
|------------------------|----------------------------|

Description

Add numeric validation to a column

Usage

```
hot_validate_character(hot, cols, choices, allowInvalid = FALSE)
```

Arguments

| | |
|--------------|---|
| hot | rhandsontable object |
| cols | vector of column names or indices |
| choices | a vector of acceptable numeric choices. It will be evaluated after min and max if specified. |
| allowInvalid | logical specifying whether invalid data will be accepted. Invalid data cells will be color red. |

See Also

[hot_validate_numeric](#)

Examples

```
library(rhandsontable)
DF = data.frame(val = 1:10, bool = TRUE, big = LETTERS[1:10],
               small = letters[1:10],
               dt = seq(from = Sys.Date(), by = "days", length.out = 10),
               stringsAsFactors = FALSE)

rhandsontable(DF) %>%
  hot_validate_character(col = "big", choices = LETTERS[1:10])
```

hot_validate_numeric *Handsontable widget*

Description

Add numeric validation to a column

Usage

```
hot_validate_numeric(hot, cols, min = NULL, max = NULL, choices = NULL,
                    exclude = NULL, allowInvalid = FALSE)
```

Arguments

| | |
|--------------|---|
| hot | rhandsontable object |
| cols | vector of column names or indices |
| min | minimum value to accept |
| max | maximum value to accept |
| choices | a vector of acceptable numeric choices. It will be evaluated after min and max if specified. |
| exclude | a vector or unacceptable numeric values |
| allowInvalid | logical specifying whether invalid data will be accepted. Invalid data cells will be color red. |

See Also

[hot_validate_character](#)

Examples

```
library(rhandsontable)
MAT = matrix(rnorm(50), nrow = 10, dimnames = list(LETTERS[1:10],
                                                  letters[1:5]))

rhandsontable(MAT * 10) %>%
  hot_validate_numeric(col = 1, min = -50, max = 50, exclude = 40)
```

```
rhandsontable(MAT * 10) %>%
  hot_validate_numeric(col = 1, choices = c(10, 20, 40))
```

renderRHandsontable *Handsontable widget*

Description

Shiny bindings for rhandsontable

Usage

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

Arguments

| | |
|--------|---|
| expr | An expression that generates threejs graphics. |
| 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. |

See Also

[rHandsontableOutput](#), [hot_to_r](#)

rhandsontable *Handsontable widget*

Description

Create a [Handsontable.js](#) widget.

Usage

```
rhandsontable(data, colHeaders, rowHeaders, useTypes = TRUE,
  readOnly = NULL, selectCallback = FALSE, width = NULL, height = NULL,
  ...)
```

Arguments

| | |
|----------------|---|
| data | a data.table, data.frame or matrix |
| colHeaders | a vector of column names. If missing colnames will be used. Setting to NULL will omit. |
| rowHeaders | a vector of row names. If missing rownames will be used. Setting to NULL will omit. |
| useTypes | logical specifying whether column classes should be mapped to equivalent Javascript types |
| readOnly | logical specifying whether the table is editable |
| selectCallback | logical enabling the afterSelect event to return data. This can be used with shiny to tie updates to a selected table cell. |
| width | numeric table width |
| height | numeric table height |
| ... | passed to hot_table |

Details

For full documentation on the package, visit <http://jrowen.github.io/rhandsontable/>

See Also

[hot_table](#), [hot_cols](#), [hot_rows](#), [hot_cell](#)

Examples

```
library(rhandsontable)
DF = data.frame(val = 1:10, bool = TRUE, big = LETTERS[1:10],
               small = letters[1:10],
               dt = seq(from = Sys.Date(), by = "days", length.out = 10),
               stringsAsFactors = FALSE)

rhandsontable(DF, rowHeaders = NULL)
```

rhandsontable-exports *rhandsontable exported operators*

Description

The following functions are imported and then re-exported from the rhandsontable package to enable use of the magrittr pipe operator with no additional library calls

`rHandsontableOutput` *Handsontable widget*

Description

Shiny bindings for rhandsontable

Usage

```
rHandsontableOutput(outputId, width = "100%", height = "100%")
```

Arguments

| | |
|---------------------------|---|
| <code>outputId</code> | output variable to read from |
| <code>width,height</code> | Must be a valid CSS unit in pixels or a number, which will be coerced to a string and have "px" appended. |

See Also

[renderRHandsontable](#)

Index

`%>%` (rhandsontable-exports), 11

`hot_cell`, 2, 4–6, 11

`hot_col`, 3, 5

`hot_cols`, 3, 4, 4, 6, 11

`hot_heatmap`, 5

`hot_rows`, 3–5, 6, 11

`hot_table`, 6, 11

`hot_to_r`, 8, 10

`hot_validate_character`, 8, 9

`hot_validate_numeric`, 8, 9

`renderRHandsontable`, 10, 12

`rhandsontable`, 7, 10

`rhandsontable-exports`, 11

`rhandsontable-package`, 2

`rHandsontableOutput`, 8, 10, 12