

Package ‘listviewer’

August 29, 2016

Type Package

Title 'htmlwidget' for Interactive Views of R Lists

Version 1.0

Date 2016-06-14

Author c(
 person(`Jos`, `de Jong`, role = c(`aut`, `cph`), comment = `jsoneditor.js library in htmlwid-
 gets/jsoneditor, http://github.com/josdejong/jsoneditor/"`),
 person(`Kenton`, `Russell`, role = c(`aut`, `cre`), comment = `R inter-
 face"`, email = `kent.russell@timelyportfolio.com"`)
)

Maintainer Kent Russell<kent.russell@timelyportfolio.com>

Description R lists especially nested lists can be very difficult to visualize or represent. 'str' just isn't enough, so this suite of 'htmlwidgets' is designed to help see, understand, and maybe even modify your R lists.

License MIT + file LICENSE

LazyData TRUE

URL <https://github.com/timelyportfolio/listviewer>

BugReports <https://github.com/timelyportfolio/listviewer/issues>

Imports htmlwidgets

Suggests htmltools, jsonlite, shiny

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-06-15 08:33:04

R topics documented:

jsonedit	2
jsonedit-shiny	3

Index	4
--------------	----------

 jsonedit

 View Lists with 'jsoneditor'

Description

jsonedit provides a flexible and helpful interactive tree-like view of lists or really any R dataset that can be represented as JSON. Eventually, this could become a very nice way to not only view but also modify R data using Shiny.

Usage

```
jsonedit(listdata = NULL, mode = "tree", modes = c("code", "form", "text",
  "tree", "view"), ..., width = NULL, height = NULL)
```

Arguments

listdata	list or String data to view. Although designed for lists, listdata can be any data source that can be rendered into JSON with jsonlite. Alternately, listdata could be a String of valid JSON. This might be helpful when dealing with an API response.
mode	string for the initial view from modes. 'tree' is the default.
modes	string c('code', 'form', 'text', 'tree', 'view') will be the default, since these are all the modes currently supported by jsoneditor.
...	list of other options for jsoneditor. This is a temporary way of trying other options in jsoneditor. In the future, this will be eliminated in favor of specific, more self-documenting and helpful arguments.
width	integer in pixels defining the width of the div container.
height	integer in pixels defining the height of the div container.

Examples

```
library(listviewer)

# using the data from the jsoneditor simple example
# in R list form
jsonedit(
  list(
    array = c(1,2,3)
    ,boolean = TRUE
    ,null = NULL
    ,number = 123
    ,object = list( a="b", c="d" )
    ,string = "Hello World"
  )
)

# jsonedit also works with a JSON string
```

```
jsonedit(  
  '{"array" : [1,2,3] , "boolean" : true, "null" : null, number = 123}'  
)  
  
# also works with most data.frames  
jsonedit( mtcars )  
  
# helpful interactive view of par  
jsonedit( par() )
```

jsonedit-shiny

Shiny Bindings for 'jsonedit'

Description

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

Usage

```
jsoneditOutput(outputId, width = "100%", height = "400px")  
  
renderJsonedit(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 jsonedit
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.

Index

`jsonedit`, [2](#)
`jsonedit-shiny`, [3](#)
`jsoneditOutput` (`jsonedit-shiny`), [3](#)
`renderJsonedit` (`jsonedit-shiny`), [3](#)