

Package: rdeck.controls (via r-universe)

September 20, 2024

Title External Interactive Controls for {rdeck} in {rmarkdown}

Version 0.2.0

Description HTML widgets for external controls for toggling layers for {rdeck} in HTML documents.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE, roclets = c("`collate", "`namespace",
`rd", `roxyglobals::global_roclet"))

RoxygenNote 7.1.1

Imports purrr, htmlwidgets, rlang, tidyselect, magrittr, rdeck (>= 0.3.0.9320), htmltools

Remotes anthonymnorth/rdeck, anthonymnorth/roxyglobals

Suggests roxyglobals (>= 0.2.1)

Repository <https://milesmbain.r-universe.dev>

RemoteUrl <https://github.com/qfes/rdeck.controls>

RemoteRef master

RemoteSha 9b2865ee82b47327c9a8bbc3f11c74c6eecdd6b8

Contents

as.tags.rdeckControls	2
rdeckControls	2
rdeckControls-shiny	3
rdeck_layer_dropdown	3

Index	5
--------------	----------

```
as.tags.rdeckControls  convert rdeckControls to html allowing !important
```

Description

convert rdeckControls to html allowing !important

Usage

```
## S3 method for class 'rdeckControls'
as.tags(x, ...)
```

Arguments

x	an rdeckControls widget
...	other args passed to htmlwidgets:::toHTML

Value

html for the widget

rdeckControls	<i>Generic rdeck control widget</i>
---------------	-------------------------------------

Description

A generic control widget constructor.

Usage

```
rdeckControls(
  targetRDeckId,
  controlType = "dropdown",
  controlData = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

rdeckControls-shiny *Shiny bindings for rdeckControls*

Description

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

Usage

```
rdeckControlsOutput(outputId, width = "100%", height = "1em")

renderRdeckControls(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 rdeckControls
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.

rdeck_layer_dropdown *Dropdown layer selector for rdeck maps*

Description

Create an external layer control for an rdeck map in rmarkdown or shiny.

Usage

```
rdeck_layer_dropdown(
  rdeck,
  layer_names,
  layer_group_names,
  initial_selection = NULL,
  label = "",
  width = "100%",
  height = "1em"
)
```

Arguments

<code>rdeck</code>	the target rdeck instance to create a control for.
<code>layer_names</code>	a tidy select expression matching layer names. e.g. <code>starts_with("demand")</code>
<code>layer_group_names</code>	a tidy select expression matching layer group names. e.g. <code>starts_with("rfs")</code>
<code>initial_selection</code>	The layer or layer group that is to be selected by default before the user has interacted with the control. Defaults to the first matched layer name, then layer group name.
<code>label</code>	Some text that immediately precedes the dropdown control.
<code>width</code>	of the control div as css dimension string.
<code>height</code>	of the control div as css dimension string.

Details

Match `layer_names` and or `layer_group_names` by using 'tidy select' syntax see [tidyselect::eval_select\(\)](#).

The `layer_names` tidyselect can only be used to toggle visibility of ungrouped layers, that is layers with no `group_name` set. This is enforced by `{rdeck}`.

Index

`as.tags.rdeckControls`, [2](#)

`rdeck_layer_dropdown`, [3](#)

`rdeckControls`, [2](#)

`rdeckControls-shiny`, [3](#)

`rdeckControlsOutput`

`(rdeckControls-shiny)`, [3](#)

`renderRdeckControls`

`(rdeckControls-shiny)`, [3](#)

`tidyselect::eval_select()`, [4](#)