Package: vueR (via r-universe)

September 25, 2024

Type Package

Title 'Vuejs' Helpers and 'Htmlwidget'

Version 0.6.0

Date 2023-10-01

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

Description Make it easy to use 'vue' in R with helper dependency functions and examples.

URL https://github.com/vue-r/vueR

BugReports https://github.com/vue-r/vueR/issues

License MIT + file LICENSE

LazyData TRUE

Imports htmltools, htmlwidgets (>= 0.6.0)

Suggests knitr, rmarkdown, shiny, testthat

RoxygenNote 7.2.3

VignetteBuilder knitr

Repository https://timelyportfolio.r-universe.dev

RemoteUrl https://github.com/vue-r/vueR

RemoteRef master

RemoteSha 61dbd730bfc2d510da0c87bccd138d862d689e7b

Contents

`html_dependency_vue` *Dependencies for Vue*

Description

Dependencies for Vue

Usage

```
html_dependency_vue(offline = TRUE, minified = TRUE)
```

Arguments

`offline`	`logical` to use local file dependencies. If `FALSE`, then the dependencies use cdn as its `src`.
`minified`	`logical` to use minified (production) version. Use `minified = FALSE` for debugging or working with Vue devtools.

Value

`htmlDependency`

See Also

Other dependencies: `html_dependency_vue3()`

Examples

```
if(interactive()){

library(vueR)
library(htmltools)

attachDependencies(
  tagList(
    tags$div(id="app","{{message}}"),
    tags$script(
    "
    var app = new Vue({
      el: '#app',
      data: {
        message: 'Hello Vue!'
      }
    });
    "
    )
  ),
  html_dependency_vue()
)
}
```

`html_dependency_vue3` *Dependencies for 'Vue3'*

Description

Dependencies for 'Vue3'

Usage

```
html_dependency_vue3(offline = TRUE, minified = TRUE)
```

Arguments

`offline`	`logical` to use local file dependencies. If `FALSE`, then the dependencies use cdn as its `src`.
`minified`	`logical` to use minified (production) version. Use `minified = FALSE` for debugging or working with Vue devtools.

Value

`htmlDependency`

See Also

Other dependencies: `html_dependency_vue()`

Examples

```
if(interactive()){

library(vueR)
library(htmltools)

browsable(
  tagList(
    tags$div(id="app","{{message}}"),
    tags$script(
    "
    var app = {
      data: function() {
        return {
          message: 'Hello Vue!'
        }
      }
    };

    Vue.createApp(app).mount('#app');
    "
    ),
```

```
    html_dependency_vue3()
  )
)
}
```

---

vue *'Vue.js' 'htmlwidget'*

Description

Use 'Vue.js' with the convenience and flexibility of 'htmlwidgets'. vue is a little different from other 'htmlwidgets' though since it requires specification of the HTML tags/elements separately.

Usage

```
vue(
  app = list(),
  width = NULL,
  height = NULL,
  elementId = NULL,
  minified = TRUE
)
```

Arguments

app	list with el and data and other pieces of a 'Vue.js' app
width, height	any valid CSS size unit, but in reality this will not currently have any impact
elementId	character id of the htmlwidget container element
minified	logical to indicate minified (minified=TRUE) or non-minified (minified=FALSE) Vue.js

Value

vue htmlwidget

See Also

Other htmlwidget: vue3()

Examples

```
if(interactive()) {

  library(vueR)
  library(htmltools)

  # recreate Hello Vue! example
  browsable(
```

```
  tagList(
    tags$div(id="app", "{{message}}"),
    vue3(
      list(
        el = "#app",
        data = list(
          message = "Hello Vue!"
        )
      )
    )
  )
)

# app2 from Vue.js introduction
browsable(
  tagList(
    tags$div(id="app-2",
      tags$span(
        "v-bind:title" = "message",
       "Hover your mouse over me for a few seconds to see my dynamically bound title!"
      )
    ),
    vue(
      list(
        el = "#app-2",
        data = list(
          message =  htmlwidgets::JS(
            "'You loaded this page on ' + new Date()"
          )
        )
      )
    )
  )
)

# app3 from Vue.js introduction
#  with a setInterval to toggle seen true and false
browsable(
  tagList(
    tags$div(id="app-3",
      tags$p("v-if"="seen", "Now you see me")
    ),
    htmlwidgets::onRender(
      vue(
        list(
          el = '#app-3',
          data = list(
            seen = TRUE
          )
        )
      ),
      "
function(el,x){
```

```
    var that = this;
    setInterval(function(){that.instance.seen=!that.instance.seen},1000);
  }
       "
     )
    )
  )

}
```

---

vue-shiny *Shiny bindings for vue*

Description

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

Output and render functions for using 'vue 3' within Shiny applications and interactive Rmd documents.

Usage

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

renderVue(expr, env = parent.frame(), quoted = FALSE)

vue3Output(outputId, width = "100%", height = "400px")

renderVue3(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 vue
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.

Examples

```
if(interactive()) {

  library(shiny)
  library(vueR)
```

```
  ui <- tagList(
    tags$div(id="app-3",
      tags$p("v-if"="seen", "Now you see me")
    ),
    vue3Output('vue1')
  )

  server <- function(input, output, session) {
    output$vue1 <- renderVue3({
      vue3(
        list(
          el = '#app-3',
          data = list(seen = TRUE),
          mounted = htmlwidgets::JS("
            function() {
              var that = this;
              setInterval(function(){that.seen=!that.seen},1000);
            }
          "),
          watch = list(
            seen = htmlwidgets::JS("function() {Shiny.setInputValue('seen',this.seen)}")
          )
        )
      )
    })

    # show that Shiny input value is being updated
    observeEvent(input$seen, {print(input$seen)})
  }

  shinyApp(ui, server)

}
```

---

vue3 *'Vue.js 3' 'htmlwidget'*

Description

Use 'Vue.js 3' with the convenience and flexibility of 'htmlwidgets'. vue3 is a little different from other 'htmlwidgets' though since it requires specification of the HTML tags/elements separately.

Usage

```
vue3(
  app = list(),
  width = NULL,
  height = NULL,
  elementId = NULL,
  minified = TRUE
)
```

Arguments

app	list with el and data and other pieces of a 'Vue.js 3' app
width, height	any valid CSS size unit, but in reality this will not currently have any impact
elementId	character id of the htmlwidget container element
minified	logical to indicate minified (minified=TRUE) or non-minified (minified=FALSE) Vue.js

Value

vue htmlwidget

See Also

Other htmlwidget: vue()

Examples

```
if(interactive()) {

  library(vueR)
  library(htmltools)

  # recreate Hello Vue! example
  browsable(
    tagList(
      tags$div(id="app", "{{message}}"),
      vue3(
        list(
          el = "#app",
          # vue 3 is more burdensome but robust requiring data as function
          #   if data is not a function then widget will auto-convert
          data = list(message = "Hello Vue3!")
          # data = htmlwidgets::JS("
          #   function() {return {message: 'Hello Vue3!'}}
          # ")
        )
      )
    )
  )

  # app2 from Vue.js introduction
  browsable(
    tagList(
      tags$div(id="app-2",
        tags$span(
          "v-bind:title" = "message",
          "Hover your mouse over me for a few seconds to see my dynamically bound title!"
        )
      ),
      vue3(
```

```
          list(
            el = "#app-2",
            # vue 3 is more burdensome but robust requiring data as function
            #   if data is not a function then widget will auto-convert
            data = htmlwidgets::JS("
              function() {
                return {message: 'You loaded this page on ' + new Date()}
              }
            ")
          )
        )
      )
    )

    # app3 from Vue.js introduction
    #  with a setInterval to toggle seen true and false
    browsable(
      tagList(
        tags$div(id="app-3",
          tags$p("v-if"="seen", "Now you see me")
        ),
        vue3(
          list(
            el = '#app-3',
            data = list(seen = TRUE),
            # data = htmlwidgets::JS("function() {return {seen: true}}"),
            mounted = htmlwidgets::JS("
              function() {
                var that = this;
                setInterval(function(){that.seen=!that.seen},1000);
              }
            ")
          )
        )
      )
    )

  }
```

Index