

# Package ‘vueR’

October 12, 2022

**Type** Package

**Title** 'Vuejs' Helpers and 'Htmlwidget'

**Version** 0.5.3

**Date** 2021-11-25

**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

**Imports** htmltools, htmlwidgets (>= 0.6.0)

**Suggests** knitr, rmarkdown, shiny, testthat

**RoxygenNote** 7.1.2

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Evan You [aut, cph] (vue library in lib, <https://vuejs.org/>; see  
BACKERS for full list of Patreon backers),  
Kent Russell [aut, cre] (R interface)

**Repository** CRAN

**Date/Publication** 2021-11-29 08:50:02 UTC

## R topics documented:

html_dependency_vue	2
vue	3
vue-shiny	5

<b>Index</b>	<b>6</b>
--------------	----------

---

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](#)

## 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()  
  )  
}
```

---

vue	<i>'Vue.js' 'htmlwidget'</i>
-----	------------------------------

---

### 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

### Examples

```
if(interactive()) {  
  
  library(vueR)  
  library(htmltools)  
  
  # recreate Hello Vue! example  
  browsable(  
    tagList(  
      tags$div(id="app", "{{message}}"),  
      vue(  
        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.

**Usage**

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

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

**Arguments**

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

# Index

[html\\_dependency\\_vue](#), 2  
[htmlDependency](#), 2  
  
[renderVue \(vue-shiny\)](#), 5  
  
[vue](#), 3  
[vue-shiny](#), 5  
[vueOutput \(vue-shiny\)](#), 5