

Package ‘aniview’

March 31, 2020

Title Animate Shiny and R Markdown Content when it Comes into View

Version 0.1.0

Description Animate Shiny and R Markdown content when it comes into view using 'animate-css' effects thanks to 'jQuery AniVar'.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports jsonlite, htmltools

Suggests shiny

URL <https://felixluginbuhl.com/aniview>,
<https://github.com/lgnbhl/aniview>

BugReports <https://github.com/lgnbhl/aniview/issues>

NeedsCompilation no

Author Félix Luginbuhl [aut, cre]

Maintainer Félix Luginbuhl <felix.luginbuhl@protonmail.ch>

Repository CRAN

Date/Publication 2020-03-31 15:20:07 UTC

R topics documented:

aniview	2
use_aniview	2

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

aniview

Animate an element when it comes into view

Description

Programmatically animate Shiny and R Markdown content when it comes into view.

Usage

```
aniview(element, animation, ...)
```

Arguments

element	An shiny or rmarkdown element.
animation	An animation from animate.css
...	Additional class element

Examples

```
library(shiny)

ui <- function(){
  fluidPage(
    use_aniview(), # add use_aniview() in the UI
    aniview(h1("Shiny with AniView"), animation = "fadeInUp"),
    aniview(textOutput("text"), animation = "zoomIn")
  )
}

server <- function(input, output, session){
  output$text <- renderText({
    print("An animated text.")
  })
}

if(interactive()) shinyApp(ui, server)
```

use_aniview*Use AniView*

Description

Use AniView

Usage

```
use_aniview(animateThreshold = 0, scrollPollInterval = 20)
```

Arguments

animateThreshold

integer. +ve numbers delay the animation sequence until the specified number of pixels have come into view. -ve numbers will trigger the animation sequence prior to the element coming into view.

scrollPollInterval

integer. frequency at which user scrolling is 'polled' i.e. tested. This is in milliseconds and is an extension to jQuery's in-built 'scroll' event/handler.

Examples

```
library(shiny)

ui <- function(){
  fluidPage(
    use_aniview(), # add use_aniview() in the UI
    aniview(h1("Shiny with AniView"), animation = "fadeInUp"),
    aniview(textOutput("text"), animation = "zoomIn")
  )
}

server <- function(input, output, session){
  output$text <- renderText({
    print("An animated text.")
  })
}

if(interactive()) shinyApp(ui, server)
```

Index

`aniview`, [2](#)

`use_aniview`, [2](#)