Package 'shinytitle'

October 14, 2022

Type Package

Title Update Browser Window Title in 'shiny' Session

Version 0.1.0

Description
Enables the ability to change or flash the title of the browser window during a 'shiny' session.

URL https://github.com/ashbaldry/shinytitle,
https://ashbaldry.github.io/shinytitle/

BugReports https://github.com/ashbaldry/shinytitle/issues

Imports shiny

Suggests testthat (>= 3.0.0)

License GPL-2

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Ashley Baldry [aut, cre]

Maintainer Ashley Baldry <arbaldry91@gmail.com>

Repository CRAN

Date/Publication 2021-06-16 10:40:02 UTC

R topics documented:

busy_window_title *Create Busy Browser Title*

Description

Change the text in the browser tab whenever the shiny application is processing any server-side code.

Usage

```
busy_window_title(title = "Running...")
```

Arguments

title	String to give the window title.

Value

The browser tab title will change whenever the shiny application is in a 'busy' state.

Note

Add use_shiny_title within the UI for busy_window_title to work.

Examples

```
if (interactive()) {
  library(shiny)

  ui <- fluidPage(
    title = "Initial Title",
    use_shiny_title(),
    busy_window_title("Sleeping..."),
    actionButton("button", "Click me for a 5 second busy title")
  )

  server <- function(input, output, session) {
    observeEvent(input$button, Sys.sleep(5))
  }

  shinyApp(ui, server)
}
```

change_window_title *Change Browser Title*

Description

Change the text that is present in the browser tab.

Usage

```
change_window_title(
  session = shiny::getDefaultReactiveDomain(),
  title = "Ready!",
  inactive_only = FALSE,
  revert_on_focus = inactive_only
)
```

Arguments

session	The session object passed to function given to shinyServer. Default is getDefaultReactiveDomain()
title	String to give the window title
inactive_only	Logical, whether or not the title should only change if the tab is not active. Default is set to FALSE
revert_on_focus	Logical, should the title revert back to the original title when the tab is in focus/active again? Only works when inactive_only = TRUE.

Value

The browser tab title will change to the new specified title.

Note

Add use_shiny_title within the UI for change_window_title to work.

Examples

```
if (interactive()) {
  library(shiny)

  ui <- fluidPage(
    title = "Initial Title",
    use_shiny_title(),
    actionButton("button", "Click me for a new title"),
    actionButton("button2", "Click me for a button when finished")
  )

  server <- function(input, output, session) {
    observeEvent(input$button, {
```

```
    change_window_title(session, "New Title")
  })

  observeEvent(input$button2, {
    Sys.sleep(3)
    change_window_title(session, "Sleep Finished", inactive_only = TRUE)
  })
}

shinyApp(ui, server)
}
```

`flashing_window_title` *Create Flashing Browser Title*

Description

Alternate the text in the browser tab between the current text and the new specified text.

Usage

```
flashing_window_title(
  session = shiny::getDefaultReactiveDomain(),
  title = "Ready!",
  inactive_only = FALSE,
  revert_on_focus = inactive_only,
  revert_on_mousemove = TRUE,
  interval = 500,
  duration = 0
)
```

Arguments

`session`	The `session` object passed to function given to `shinyServer`. Default is `getDefaultReactiveDomain()`
`title`	String to give the window title.
`inactive_only`	Logical, whether or not the title should only change if the tab is not active. Default is set to `FALSE`.
`revert_on_focus`	Logical, should the title revert back to the original title when the tab is in focus/active again? Only works when `inactive_only = TRUE`.
`revert_on_mousemove`	Logical, should the title revert back to the original title when the mouse is moved in the tab? Default is set to `TRUE`.
`interval`	Time (in milliseconds) to flip between the original title and the new title.
`duration`	Time (in milliseconds) to stop flashing the title. 0 (the default) means it will flash indefinitely.

Value

The browser tab title will change between the original and newly specified title.

Note

Add `use_shiny_title` within the UI for `flashing_window_title` to work.

Examples

```
if (interactive()) {
  library(shiny)

  ui <- fluidPage(
    title = "Initial Title",
    use_shiny_title(),
    actionButton("button", "Click me for a 10 second flashing title"),
    actionButton("button2", "Click me for a delayed flashing button")
  )

  server <- function(input, output, session) {
    observeEvent(input$button, {
      flashing_window_title(
        session, "--- Flash ---", revert_on_mousemove = FALSE, duration = 10000
      )
    })

    observeEvent(input$button2, {
      Sys.sleep(3)
      flashing_window_title(
        session, "Please Come Back", inactive_only = TRUE, interval = 1000
      )
    })
  }

  shinyApp(ui, server)
}
```

`run_shinytitle_example`

Run shinytitle Example

Description

Launch one of the examples contained in the `shinytitle` package:

- toggleAn example showing the effects of simple change to the title and flashing title
- busyAn example of when the title changes when the shiny app is busy running calculations

Usage

```
run_shinytitle_example(example = c("toggle", "busy"), ...)
```

Arguments

example	Choose between toggle and busy
...	other arguments sent to runApp

Examples

```
if (interactive()) {
  library(shiny)
  run_shinytitle_example()
}
```

---

use_shiny_title *Enable* shinytitle

Description

Add this function to the UI of a shiny application in order for you to be able to update the browser title.

Usage

```
use_shiny_title()
```

Value

A script tag that enables shinytitle to work within a shiny app.

See Also

change_window_title flashing_window_title

Examples

```
if (interactive()) {
  library(shiny)

  ui <- fluidPage(
    title = "Initial Title",
    use_shiny_title(),
    actionButton("button", "Click me for a new title"),
  )

  server <- function(input, output, session) {
```

```
    observeEvent(input$button, {
      change_window_title(session, "New Title")
    })
  }

  shinyApp(ui, server)
}
```

Index