

# Package ‘lareshiny’

July 22, 2025

**Type** Package

**Title** Lares 'shiny' Modules

**Version** 0.0.3

**Maintainer** Bernardo Lares <laresbernardo@gmail.com>

**Description** Useful 'shiny' production-ready modules and helpers such as login window and visualization tools.

**License** AGPL-3

**Encoding** UTF-8

**Imports** dplyr, DT, htmltools, miniUI, shiny, shinydashboard, shinyWidgets

**Suggests** h2o, lares

**URL** <https://github.com/laresbernardo/lareshiny>

**BugReports** <https://github.com/laresbernardo/lareshiny/issues>

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Bernardo Lares [cre, aut]

**Repository** CRAN

**Date/Publication** 2023-07-06 19:00:02 UTC

## Contents

autoline_shiny . . . . .	2
custom_footer . . . . .	2
custom_header . . . . .	3
js_op . . . . .	4
js_op_aux . . . . .	5
lareshiny . . . . .	5
model_exporter . . . . .	6
module_login . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

---

autoline_shiny	<i>Automatically wrap ggplot2 texts depending on rendered size</i>
----------------	--------------------------------------------------------------------

---

### Description

This function automatically adapts the length of a title or subtitle text so that it wraps long texts in separate lines. This is useful to avoid trimming long texts when the rendered shiny size is shorter.

### Usage

```
autoline_shiny(session, text, plot_name, font_size = 14, n = 1.6)
```

### Arguments

session	Shiny's session object.
text	Character. Text for the title or subtitle or any text to be wrapped.
plot_name	Character. The name of the output plot.
font_size, n	Numeric. Values to help calculate a proxy of length.

### Value

Character. String with wrapped values.

---

custom_footer	<i>Custom Footer: Customizes for Shiny's Footer</i>
---------------	-----------------------------------------------------

---

### Description

This function personalizes your Shiny dashboard's footer with your company's name, link, help mail.

### Usage

```
custom_footer(  
  company = "MyCoolCompany",  
  text = "ShinyDashboard",  
  site = "https://github.com/laresbernardo/lareshiny",  
  helpmail = "myemail@mydomain.com",  
  lang = "en"  
)
```

**Arguments**

company, text, site	Character. Your company's name and URL and the project's name to show in footer.
helpmail	Character. Developers mail
lang	Character. Language for the text.

**Value**

HTML string (tags\$footer).

---

custom_header	<i>Custom Header: Customized for Shiny's Header</i>
---------------	-----------------------------------------------------

---

**Description**

This function personalizes your Shiny dashboard's header with logo, links, favicon, font, and texts.

**Usage**

```
custom_header(
  title = "MyLareShiny",
  site = NULL,
  favicon = NULL,
  font = "Montserrat",
  logosrc = NULL,
  logo_height = "40px",
  logo_width = NULL,
  loadingsrc = NULL,
  load_height = "40px",
  load_width = NULL,
  text = Sys.Date(),
  type = 1
)
```

**Arguments**

title	Character. Your 'shiny' app title (displayed in Navigators' tab)
site	Character. URL for your site (opens when logo is clicked)
favicon	Character. Image for your favicon. Save file in www directory.
font	Character. Font for whole 'shiny' app. Use Google Fonts names.
logosrc, loadingsrc	Character. Logo image and loading image. For local files, save them in www directory to call them directly.

logo\_height, logo\_width, load\_height, load\_width  
 Integer. Logo and loading images dimensions.  
 text  
 Character. Text displayed in top right corner.  
 type  
 Integer. 1 for complete dashboardHeader results, 2 for title results.

**Value**

A dashboardHeader object.

**Examples**

```
if (interactive()) {
  shinyApp(
    ui = dashboardPage(
      custom_header(
        title = "MyLareShiny",
        site = "https://github.com/laresbernardo/lareshiny",
        font = "Montserrat",
        logosrc = "logo.png",
        text = Sys.Date()
      ),
      dashboardSidebar(disable = TRUE),
      dashboardBody()
    ),
    server = function(input, output) { }
  )
}
```

---

 js\_op

*Totals Row for datatables*


---

**Description**

Totals Row for datatables

**Usage**

```
js_op(column, operation, txt = "", signif = 3)
```

**Arguments**

column  
 Integer. Starting from 0, which column to operate  
 operation  
 Character. Select from sum, mean, count, custom  
 txt  
 Character. Insert text before (or instead) operation  
 signif  
 Integer. How many decimals to consider when operating

**Value**

Text that can be parsed as javascript.

---

`js_op_aux`*Helper for javascript code*

---

**Description**

Helper for javascript code

**Usage**

```
js_op_aux(type, df = NULL)
```

**Arguments**

`type` Character. Select start, end or sketch.  
`df` data.frame. Needed for sketch only.

**Value**

Text that can be parsed as javascript.

---

`lareshiny`*Lares Shiny Modules*

---

**Description**

R library for quick useful Shiny modules

**Author(s)**

Bernardo Lares (laresbernardo@gmail.com)

**See Also**

Useful links:

- <https://github.com/laresbernardo/lareshiny>
- Report bugs at <https://github.com/laresbernardo/lareshiny/issues>

---

model_exporter	<i>Export Results from h2o_automl()</i>
----------------	-----------------------------------------

---

**Description**

Friendly app that helps you export lares' h2o\_automl() results

**Usage**

```
model_exporter(model)
```

**Arguments**

model	List. Result from h2o_automl()
-------	--------------------------------

**Value**

A shiny interactive app

**Examples**

```
if (interactive()) {
  model_exporter(model)
}
```

---

module_login	<i>Login Module for Shiny</i>
--------------	-------------------------------

---

**Description**

Login Module for Shiny with local User and Password. This must be added in the server section (no need to add UI)

**Usage**

```
module_login(
  input,
  session,
  users = c("123", "321"),
  pwds = c("123", "321"),
  logo = NA,
  logo_height = "100px",
  lang = "es",
  style = list(botton_txt_colour = "#FFFFFF", botton_bgd_colour = "#EBB600"),
  change_text = list(),
  logged = FALSE,
  personal = "MacBookBLB.local"
)
```

**Arguments**

input, session	Shiny's input and session objects.
users, pwds	Character Vector. User and password combinations accepted
logo	Character. Select image for logo display. Host local file in your www directory preferably
logo_height	Character. Height for rendering the logo.
lang	Character. Language. Currently accepted: es, en
style	List. Possible values for styling the module such as botton_txt_colour and botton_bgd_colour.
change_text	Named list. Change the default texts used.
logged	Boolean. You might want to set to TRUE when developing or testing your app so this module doesn't show up every time. Check the personal parameter as well
personal	Character vector. If you wish to auto-login certain user(s), set the values from Sys.info()[["nodename"]].

**Value**

A reactiveValues object

**Examples**

```
if (interactive()) {  
  ui <- fluidPage("Hello lareshiny!")  
  server <- function(input, output, session) {  
    login <- module_login(input, session, personal = "")  
    observe({  
      if (login$authenticated) message("We are in!")  
    })  
  }  
  shinyApp(ui, server)  
}
```

# Index

[autoline\\_shiny](#), [2](#)

[custom\\_footer](#), [2](#)

[custom\\_header](#), [3](#)

[js\\_op](#), [4](#)

[js\\_op\\_aux](#), [5](#)

[lareshiny](#), [5](#)

[lareshiny-package \(lareshiny\)](#), [5](#)

[model\\_exporter](#), [6](#)

[module\\_login](#), [6](#)