

# Package: litter (via r-universe)

September 1, 2024

**Title** 'Lit' Components for 'Shiny'

**Version** 0.0.1.9000

**Description** 'Lit' components for 'Shiny'.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Imports** shiny, jsonlite, htmltools

**Suggests** packer

**Repository** <https://devopifex.r-universe.dev>

**RemoteUrl** <https://github.com/devOpifex/litter>

**RemoteRef** HEAD

**RemoteSha** 0036013e533bce989144f7e564c0066ec11699a0

## Contents

gallery . . . . .	2
litActionButton . . . . .	2
litActionLink . . . . .	3
litCheckboxesInput . . . . .	4
litColorInput . . . . .	5
litDatalistInput . . . . .	6
litFilterInput . . . . .	7
litPasswordInput . . . . .	7
litRadioInput . . . . .	8
litRangeInput . . . . .	9
litSelectInput . . . . .	10
litSelectizeInput . . . . .	11
litSwitchInput . . . . .	11
litTextAreaInput . . . . .	12
litTextInput . . . . .	13
litTextLabelInput . . . . .	14

litTogglerInput . . . . .	15
lit_default_color . . . . .	15
update_input . . . . .	16

<b>Index</b>	<b>18</b>
--------------	-----------

---

gallery	<i>Gallery</i>
---------	----------------

---

### Description

Demo of inputs.

### Usage

```
gallery()
```

---

litActionButton	<i>Lit Action Button</i>
-----------------	--------------------------

---

### Description

Create an action button.

### Usage

```
litActionButton(  
  name = "",  
  content = "Click me",  
  ...,  
  id = NULL,  
  class = NULL,  
  value = NULL,  
  callback = NULL,  
  send_on_render = FALSE  
)
```

### Arguments

name	Name of the input.
content	Content of the button, a character string or valid shiny::tags.
...	Passed to props.
id	Id of the input.
class	Any additional classes.
value	Value of the input.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

## Examples

```
library(shiny)

ui <- fluidPage(
  litActionButton(
    "btn",
    "First btn"
  ),
  litActionButton(
    "btn",
    "Second btn"
  )
)

server <- function(input, output, session){
  observeEvent(input$btn, {
    print(input$btn)
  })
}

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

---

litActionLink

*Lit Action Link*

---

## Description

Create an action button.

## Usage

```
litActionLink(
  name = "",
  content = "Click me",
  ...,
  id = NULL,
  class = NULL,
  value = NULL,
  callback = NULL,
  send_on_render = FALSE
)
```

## Arguments

name	Name of the input.
content	Content of the button, a character string or valid shiny::tags.

...	Passed to props.
id	Id of the input.
class	Any additional classes.
value	Value of the input.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

**Examples**

```
library(shiny)

ui <- fluidPage(
  litActionLink(
    "btn",
    "First btn"
  ),
  litActionLink(
    "btn",
    "Second btn"
  )
)

server <- function(input, output, session){
  observeEvent(input$btn, {
    print(input$btn)
  })
}

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

---

`litCheckboxesInput`      *Lit Checkboxes Input*

---

**Description**

Create checkboxes input.

**Usage**

```
litCheckboxesInput(
  name = "",
  choices = list(),
  ...,
  id = NULL,
  value = list(),
```

```

    class = NULL,
    callback = NULL,
    send_on_render = TRUE
  )

```

### Arguments

name	Name of the input.
choices	List of options.
...	Passed to props.
id	Id of the input.
value	choices that are checked.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

---

litColorInput	<i>Lit Color Input</i>
---------------	------------------------

---

### Description

Create a color input.

### Usage

```

litColorInput(
  name = "",
  value = "",
  placeholder = "",
  ...,
  id = NULL,
  class = NULL,
  callback = NULL,
  send_on_render = TRUE
)

```

### Arguments

name	Name of the input.
value	Value of the input.
placeholder	Placeholder text.
...	Passed to props.
id	Id of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render

---

litDatalistInput      *Lit Datalist Input*

---

### Description

Create a datalist input.

### Usage

```
litDatalistInput(  
  name = "",  
  choices = list(),  
  ...,  
  id = NULL,  
  value = NULL,  
  class = NULL,  
  placeholder = NULL,  
  callback = NULL,  
  send_on_render = TRUE  
)
```

### Arguments

name	Name of the input.
choices	List of options.
...	Passed to props.
id	Id of the input.
value	Value of the input.
class	Any additional classes.
placeholder	Placeholder value.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

### Details

Can be server-side rendered with [update\\_datalist\\_input\(\)](#).

### See Also

[update\\_datalist\\_input\(\)](#)

---

litFilterInput	<i>Lit Filter Input</i>
----------------	-------------------------

---

### Description

Create a filter input.

### Usage

```
litFilterInput(  
  name = "",  
  ...,  
  dataset = NULL,  
  id = NULL,  
  value = NULL,  
  class = NULL,  
  callback = NULL,  
  send_on_render = TRUE  
)
```

### Arguments

name	Name of the input.
...	Passed to props.
dataset	Dataset to retrieve variables.
id	Id of the input.
value	Value of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

---

litPasswordInput	<i>Lit Password Input</i>
------------------	---------------------------

---

### Description

Create a password input.

**Usage**

```
litPasswordInput(
  name = "",
  value = "",
  placeholder = "",
  ...,
  id = NULL,
  class = NULL,
  callback = NULL,
  send_on_render = TRUE,
  send_on = c("deferred", "enter")
)
```

**Arguments**

name	Name of the input.
value	Value of the input.
placeholder	Placeholder text.
...	Passed to props.
id	Id of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.
send_on	Whether to send the input when the user hits the enter key or to send it in a deferred manner.

---

 litRadioInput

*Lit Radio Input*


---

**Description**

Create radio input.

**Usage**

```
litRadioInput(
  name = "",
  choices = list(),
  ...,
  id = NULL,
  value = NULL,
  class = NULL,
  callback = NULL,
  send_on_render = TRUE
)
```



**Arguments**

name	Name of the input.
choices	List of options.
...	Passed to props.
id	Id of the input.
value	choices that are checked.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

---

litRangeInput	<i>Lit Range Input</i>
---------------	------------------------

---

**Description**

Create a range input.

**Usage**

```
litRangeInput(
  name = "",
  ...,
  id = NULL,
  min = 0L,
  max = 10L,
  step = NULL,
  value = NULL,
  class = NULL,
  callback = NULL,
  send_on_render = TRUE
)
```

**Arguments**

name	Name of the input.
...	Passed to props.
id	Id of the input.
min, max, value	Minimum, maximum, and initial value of the range.
step	Incremental steps the range must take.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

**Examples**

```

library(shiny)

ui <- fluidPage(
  litRangeInput("range")
)

server <- function(input, output, session){
  observeEvent(input$range, {
    print(input$range)
  })
}

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

```

litSelectInput

*Lit Select Input***Description**

Create a select input.

**Usage**

```

litSelectInput(
  name = "",
  choices = list(),
  ...,
  id = NULL,
  value = NULL,
  class = NULL,
  callback = NULL,
  send_on_render = TRUE
)

```

**Arguments**

name	Name of the input.
choices	List of options.
...	Passed to props.
id	Id of the input.
value	Value of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

---

litSelectizeInput      *Lit Selectize Input*

---

### Description

Create a selectize input.

### Usage

```
litSelectizeInput(  
  name = "",  
  choices = list(),  
  ...,  
  id = NULL,  
  value = list(),  
  class = NULL,  
  callback = NULL,  
  send_on_render = TRUE  
)
```

### Arguments

name	Name of the input.
choices	List of options.
...	Passed to props.
id	Id of the input.
value	Value of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

---

litSwitchInput      *Lit Switch Input*

---

### Description

Create a switch input.

**Usage**

```
litSwitchInput(  
  name = "",  
  value = TRUE,  
  label = "",  
  ...,  
  id = NULL,  
  class = NULL,  
  callback = NULL,  
  send_on_render = TRUE  
)
```

**Arguments**

name	Name of the input.
value	Value of the input.
label	Label of the button.
...	Passed to props.
id	Id of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

---

litTextAreaInput	<i>Lit Text Area Input</i>
------------------	----------------------------

---

**Description**

Create a text area input.

**Usage**

```
litTextAreaInput(  
  name = "",  
  value = "",  
  placeholder = "",  
  ...,  
  id = NULL,  
  class = NULL,  
  callback = NULL,  
  send_on_render = TRUE  
)
```

**Arguments**

name	Name of the input.
value	Value of the input.
placeholder	Placeholder text.
...	Passed to props.
id	Id of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on render.

---

litTextInput	<i>Lit Text Input</i>
--------------	-----------------------

---

**Description**

Create a text input.

**Usage**

```
litTextInput(
  name = "",
  value = "",
  placeholder = "",
  ...,
  id = NULL,
  class = NULL,
  callback = NULL,
  send_on_render = TRUE,
  send_on = c("deferred", "enter")
)
```

**Arguments**

name	Name of the input.
value	Value of the input.
placeholder	Placeholder text.
...	Passed to props.
id	Id of the input.
class	Any additional classes.
callback	A JavaScript callback functions that accepts one argument.
send_on_render	Whether to send the input value on connect.
send_on	Whether to send the input when the user hits the enter key or to send it in a deferred manner.

---

`litTextLabelInput`      *Lit Text Label Input*

---

### **Description**

Create a text input with a floating label.

### **Usage**

```
litTextLabelInput(  
  name = "",  
  label,  
  value = "",  
  placeholder = label,  
  ...,  
  id = NULL,  
  class = NULL,  
  callback = NULL,  
  send_on_render = TRUE,  
  send_on = c("deferred", "enter")  
)
```

### **Arguments**

<code>name</code>	Name of the input.
<code>label</code>	Label of the input.
<code>value</code>	Value of the input.
<code>placeholder</code>	Placeholder text.
<code>...</code>	Passed to props.
<code>id</code>	Id of the input.
<code>class</code>	Any additional classes.
<code>callback</code>	A JavaScript callback functions that accepts one argument.
<code>send_on_render</code>	Whether to send the input value on render.
<code>send_on</code>	Whether to send the input when the user hits the enter key or to send it in a deferred manner.

---

litTogglerInput	<i>Lit Toggler Input</i>
-----------------	--------------------------

---

### Description

Create an input that is initially hidden but revealed on click. This hijacks the original input which no longer sends values to the server.

### Usage

```
litTogglerInput(name, display, input, ..., callback = NULL, restore = TRUE)
```

### Arguments

name	Name of the input.
display	The content to display.
input	The input to reveal.
...	Passed to props.
callback	A JavaScript callback functions that accepts one argument.
restore	Whether to set the input value to the display text and vice versa.

### Details

This can be updated with the feedback argument in [update\\_input\(\)](#) as well as `accept`, a boolean to indicate whether to accept the value set.

---

lit_default_color	<i>Default Global Color</i>
-------------------	-----------------------------

---

### Description

Default Global Color

### Usage

```
lit_default_color()
```

---

`update_input`*Update inputs*

---

### Description

Update inputs.

### Usage

```
update_input(  
  ...,  
  name = NULL,  
  id = NULL,  
  selector = NULL,  
  session = shiny::getDefaultReactiveDomain()  
)
```

```
update_datalist_input(  
  choices,  
  ...,  
  name = NULL,  
  id = NULL,  
  selector = NULL,  
  session = shiny::getDefaultReactiveDomain()  
)
```

```
update_selectize_input(  
  choices,  
  ...,  
  name = NULL,  
  id = NULL,  
  selector = NULL,  
  session = shiny::getDefaultReactiveDomain()  
)
```

```
update_filter_input(  
  dataset,  
  ...,  
  name = NULL,  
  id = NULL,  
  selector = NULL,  
  session = shiny::getDefaultReactiveDomain()  
)
```

### Arguments

... Props to update, e.g.: value.



<code>name</code>	Name of the input.
<code>id</code>	Id of the input.
<code>selector</code>	A CSS selector.
<code>session</code>	A Shiny session.
<code>choices</code>	Vector (optionally named) of options.
<code>dataset</code>	Dataset to update <a href="#">litDatalistInput()</a>

**Details**

Set either of `name`, `id`, or `selector`. The function [update\\_datalist\\_input\(\)](#) is used to server-side render the options.

# Index

gallery, [2](#)

lit\_default\_color, [15](#)

litActionButton, [2](#)

litActionLink, [3](#)

litCheckboxesInput, [4](#)

litColorInput, [5](#)

litDatalistInput, [6](#)

litDatalistInput(), [17](#)

litFilterInput, [7](#)

litPasswordInput, [7](#)

litRadioInput, [8](#)

litRangeInput, [9](#)

litSelectInput, [10](#)

litSelectizeInput, [11](#)

litSwitchInput, [11](#)

litTextAreaInput, [12](#)

litTextInput, [13](#)

litTextLabelInput, [14](#)

litTogglerInput, [15](#)

update\_datalist\_input (update\_input), [16](#)

update\_datalist\_input(), [6](#), [17](#)

update\_filter\_input (update\_input), [16](#)

update\_input, [16](#)

update\_input(), [15](#)

update\_selectize\_input (update\_input),  
[16](#)