

# Package ‘spoiler’

October 14, 2022

**Type** Package

**Title** Blur 'HTML' Elements in 'Shiny' Applications Using  
'Spoiler-Alert.js'

**Version** 1.0.0

**Description** It can be useful to temporarily hide some text or other HTML elements in 'Shiny' applications. Building on 'Spoiler-Alert.js', it is possible to select the elements to hide at startup, to partially reveal them by hovering them, and to completely show them when clicking on them.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** htmltools, shiny

**URL** <https://github.com/etiennebacher/spoiler>

**BugReports** <https://github.com/etiennebacher/spoiler/issues>

**RoxygenNote** 7.1.1

**Suggests** spelling

**Language** en-US

**NeedsCompilation** no

**Author** Etienne Bacher [aut, cre, cph]

**Maintainer** Etienne Bacher <etienne.bacher@protonmail.com>

**Repository** CRAN

**Date/Publication** 2021-06-07 07:40:02 UTC

## R topics documented:

use\_spoiler . . . . . 2

**Index** . . . . . 3

---

`use_spoiler`*Use Spoiler-alert*

---

**Description**

Load Spoiler-alert dependencies, mark some elements as spoilers, and define the blurring intensity for these elements.

**Usage**

```
use_spoiler()

spoiler(ui_element)

hide_spoilers(max = 10, partial = 4)
```

**Arguments**

<code>ui_element</code>	Element to blur
<code>max</code>	Blurring intensity of the element by default
<code>partial</code>	Blurring intensity when hovering the element

**Examples**

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

  # works better in browser

  ui <- fluidPage(
    # Load dependencies
    use_spoiler(),
    column(
      3,
      p("this text can be seen"),
      spoiler(p("this one can't be seen")) # This text will be blurred
    ),
    hide_spoilers()
  )

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

  shinyApp(ui, server)
}
```

# Index

`hide_spoilers (use_spoiler), 2`

`spoiler (use_spoiler), 2`

`use_spoiler, 2`