

# Package: shinyReports (via r-universe)

June 29, 2026

**Title** R Markdown HTML Tab Rendering for Shiny

**Version** 1.0.3

**Description** Render R Markdown reports to HTML and display them instantly in a new browser tab from within a 'shiny' application. Replaces file downloads with a seamless in-browser approach.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Imports** rmarkdown, shiny

**Suggests** htmltools, knitr, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Alexander Sadler [aut, cre], Kirsty Twigger [ctb]

**Maintainer** Alexander Sadler <alexander@apexomic.com>

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

**Date/Publication** 2026-06-29 14:40:02 UTC

**RemoteUrl** <https://github.com/cran/shinyReports>

**RemoteRef** HEAD

**RemoteSha** 6e16b4005cec4470bd4b99c4ebed8cb7156db0db

## Contents

renderReport . . . . .	2
reportButton . . . . .	3
<b>Index</b>	<b>4</b>

---

renderReport	<i>Render report to a new window</i>
--------------	--------------------------------------

---

### Description

Server-side logic to compile .Rmd and send HTML to client.

### Usage

```
renderReport(  
  inputId,  
  title = "Report",  
  rmd_file,  
  params = NULL,  
  envir = parent.frame()  
)
```

### Arguments

inputId	The input slot that will be used to access the value.
title	The title to label the report tab.
rmd_file	The input file to be rendered (.Rmd).
params	A list of named parameters that override custom params specified within the YAML front-matter.
envir	The environment in which the code chunks are to be evaluated during knitting.

### Value

No return value, called for side effects. Registers a `shiny::observeEvent` that listens for the specified `inputId` button click and renders the given .Rmd file to a temporary HTML file, the content of which is sent to the client via a custom 'shiny' message ("openWindow") to open a new browser tab/window. If the render fails, a 'shiny' error notification is displayed.

### See Also

[reportButton\(\)](#)

### Examples

```
if (interactive()) {  
  server <- function(input, output, session) {  
    renderReport(  
      inputId = "knit_report",  
      title = "My Report",  
      rmd_file = "report.Rmd",  
      params = list(date = Sys.Date())  
    )  
  }  
}
```

```

  }
}
```

---

reportButton	<i>Report download button</i>
--------------	-------------------------------

---

## Description

Modifies `shiny::actionButton()` to generate and open an R Markdown report in a new window tab.

## Usage

```
reportButton(inputId, label = "View Report", icon = NULL, ...)
```

## Arguments

<code>inputId</code>	The input slot that will be used to access the value.
<code>label</code>	The contents of the button or link—usually a text label, but you could also use any other HTML, like an image.
<code>icon</code>	An optional <code>icon()</code> to appear on the button.
<code>...</code>	Named attributes to be applied to the button or link.

## Value

A `shiny.tag.list` object definition (from `tagList`) containing two components:

- A `singleton` `<script>` tag, injected into the page `<head>`, that registers a custom 'shiny' message handler ("openWindow") for opening the rendered report in a new browser tab/window.
- A `actionButton` with the argument definitions of `inputId`, `label` and `icon`.

The return value should only be included under a `fluidPage` 'shiny' definition.

## See Also

[renderReport\(\)](#)

## Examples

```

if (interactive()) {
  ui <- fluidPage(
    reportButton(
      inputId = "knit_report",
      label = "Generate Report"
    )
  )
}
```

# Index

`actionButton`, [3](#)

`fluidPage`, [3](#)

`icon()`, [3](#)

`renderReport`, [2](#)

`renderReport()`, [3](#)

`reportButton`, [3](#)

`reportButton()`, [2](#)

`shiny::actionButton()`, [3](#)

`singleton`, [3](#)

`tagList`, [3](#)