

Package: shinyhelper (via r-universe)

August 31, 2024

Type Package

Title Easily Add Markdown Help Files to 'shiny' App Elements

Version 0.3.2.9000

BugReports <https://github.com/cwthom/shinyhelper/issues>

Description Creates a lightweight way to add markdown helpfiles to 'shiny' apps, using modal dialog boxes, with no need to observe each help button separately.

License GPL-3

Imports shiny, markdown

RoxygenNote 6.1.1

Encoding UTF-8

Repository <https://cwthom.r-universe.dev>

RemoteUrl <https://github.com/cwthom/shinyhelper>

RemoteRef HEAD

RemoteSha 7f5b4806102ec02374d293db84d1aa913d0e2641

Contents

create_help_files	2
helper	2
observe_helpers	3
shinyhelper	4
shinyhelper_demo	4

Index	6
--------------	----------

`create_help_files` *Create Directory of Help Files*

Description

Helper function to be run interactively - it will initialise a directory of help files (.md files) in `help_dir`, using the vector of files. Pass it a vector of filenames which you wish to use with `type = "markdown"` in your calls to [helper](#).

Usage

```
create_help_files(files, help_dir = "helpfiles")
```

Arguments

`files` A character vector of names to use in creating help files.
`help_dir` A character string of the directory to use for help files.

Examples

```
create_help_files(files = c("Clusters", "Columns", "PlotHelp"),
                 help_dir = "helpfiles")
```

`helper` *Augment a shiny.tag with a question mark helper button*

Description

Add an action button to your shiny UI with a question mark help icon (by default). In combination with [observe_helpers](#) this icon will produce a modal dialog box in the app, with content and title as specified. You may pass inline content for the modal, or else specify the name of a markdown file to include.

Usage

```
helper(shiny_tag, icon = "question-circle", colour = NULL,
       type = "markdown", title = "", content = "", size = "m",
       buttonLabel = "Okay", easyClose = TRUE, fade = FALSE, ...)
```

Arguments

shiny_tag	A shiny element, such as an input or output (but any shiny tag will do).
icon	A character string of the name of the icon to display (font awesome).
colour	Any valid CSS colour for the icon.
type	Either 'markdown' to include a helpfile, or 'inline' to specify text.
title	The title to use for your modal. Leave as "" to have no title (for example, if your markdown document has a header you wish to use instead).
content	If type = "markdown", the name of your markdown file; if type = "inline", a character vector of text/HTML to include. This will be pasted together with line breaks. You need not add '.md' to the markdown filename.
size	Either 's', 'm' or 'l' - the size of the modal dialog to display.
buttonLabel	The text for the modal button - "Ok" by default.
easyClose	Whether the modal can be dismissed by pressing Escape, or clicking outside the modal. Defaults to TRUE.
fade	Whether the modal has a fade in animation. Defaults to FALSE.
...	Other arguments to pass to the div containing the icon.

Examples

```
helper(shiny::actionButton("go", "click me!"),
       icon = "exclamation",
       colour = "red",
       type = "markdown",
       content = "ClickHelp") # looks for 'helpfiles/ClickHelp.md'
```

observe_helpers *Observe Helper Action Buttons*

Description

Function to show a modal dialog, observing each of the help icons in the app.

Usage

```
observe_helpers(session = shiny::getDefaultReactiveDomain(),
               help_dir = "helpfiles", withMathJax = FALSE)
```

Arguments

session	The session object in your shiny app.
help_dir	A character string of the directory containing your helpfiles.
withMathJax	If TRUE the MathJax library is added to allow for math expressions in markdown content.

Examples

```
server <- function(input, output, session){  
  
  # use anywhere in your server.R script  
  observe_helpers()  
  
  # rest of server.R  
  # ...  
  # ...  
}
```

shinyhelper

shinyhelper.

Description

Easily add markdown helpfiles to your shiny apps, by adding a single function around a shiny app element. This adds a small icon (a question mark, by default) which presents content in a modal dialog box when clicked. The content can be passed inline or given in a markdown file.

Details

There is also a convenient function to initialise a directory of markdown files given a vector of filenames.

A demo is available at <https://cwthom94.shinyapps.io/shinyhelper-demo/>

For further reference on how to use the package, please refer to <https://github.com/cwthom/shinyhelper>

shinyhelper_demo

Launch the Example App

Description

An example app demonstrating the use of the shinyhelper package.

Usage

```
shinyhelper_demo()
```

Examples

```
## Not run:  
if (interactive()) {  
  shinyhelper_demo()  
}  
  
## End(Not run)
```

Index

`create_help_files`, 2

`helper`, 2, 2

`observe_helpers`, 2, 3

`shinyhelper`, 4

`shinyhelper-package` (`shinyhelper`), 4

`shinyhelper_demo`, 4