Package 'roxy.shinylive'

November 21, 2024

Type Package

Title A 'roxygen2' Extension for 'Shinylive'

Version 1.0.0

Description An extension for 'roxygen2' to embed 'Shinylive' applications
in the package documentation.

License Apache License 2.0

URL https://github.com/insightsengineering/roxy.shinylive/

BugReports https://github.com/insightsengineering/roxy.shinylive/issues

Depends R (>= 4.0)

Imports glue, jsonlite (>= 1.8.6), lzstring (>= 0.1.3), roxygen2 (>=
7.2.0), stringr (>= 0.4)

Suggests pkgdown (>= 2.0.0), testthat (>= 3.1.5), withr (>= 2.4.3)

Config/Needs/verdepcheck tidyverse/glue, jeroen/jsonlite,
lzstring=parmsam/lzstring-r, r-lib/roxygen2, tidyverse/stringr,
r-lib/pkgdown, r-lib/testthat, r-lib/withr

Config/Needs/website insightsengineering/nesttemplate

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

Collate 'tag_examplesShinylive.R' 'parse_url.R'

NeedsCompilation no

Author Pawel Rucki [aut, cre],
F. Hoffmann-La Roche AG [cph, fnd]

Maintainer Pawel Rucki <pawel.rucki@roche.com>

Repository CRAN

Date/Publication 2024-11-21 17:40:06 UTC

Contents

---

create_shinylive_url   *Creates Shinylive url based on app code.*

Description

Creates Shinylive url based on app code.

Usage

```
create_shinylive_url(code, mode = c("app", "editor"), header = TRUE)
```

Arguments

code	(character(1)) A string with app code.
mode	(character(1)) A string with mode. One of "app" or "editor". Default is "app".
header	(logical(1)) A logical value indicating whether to include header. Ignored if mode is "editor".

Value

(character(1)) Shinylive app url.

Examples

```
code <- "this is your app code as a string"
create_shinylive_url(code)
create_shinylive_url(code, header = FALSE)
create_shinylive_url(code, mode = "editor")
```

---

tag-examplesShinylive   *Custom* `@examplesShinylive` *tag.*

Description

This function generates a new "Examples in Shinylive" section in the documentation. This section contains URL to the application in Shinylive and for HTML outputs: an iframe with the application. If no code is provided then the code is taken from the following @examples or @examplesIf tag.

Usage

```
#' @examplesShinylive${1:# example code (optional)}
```

Details

The application code must be executable inside Shinylive. If the application code includes functions from your package, you must add `library(<package>)` beforehand. For more information, refer to the Decoration section on how to use and decorate existing examples.

Note: All the packages used in the application code need to be installable in WebR. See this article for more details.

Decoration

To avoid repetition between the `@examplesShinylive` and `@examples` sections contents, there are special string literals to be used inside `@examplesShinylive` tag content that allow you to access the content(s) of the `@examples` or `@examplesIf` tags. These literals should be used as expressions embraced with {{ }}, which are then interpolated using `glue::glue_data(..., .open = "{{"`, `.close = "}}")`.

The following keywords are available:

- "{{ next_example }}" - (the default if empty) "raw" element of the next example
- "{{ prev_example }}" - "raw" element of the previous example
- "{{ tags_examples }}" - a list of @examples or @examplesIf tags
- "{{ examples }}" - a list of "raw" elements from `tags_examples` list elements

This allows you to access and decorate existing example code to create executable application code for Shinylive. Refer to the examples section for possible use cases.

Examples

```
# As a part of documentation:

# basic example:
#' (docs)
#' @examplesShinylive
#' @examples
#' (example code)
```

```
# using keywords - `{{ next_example }}`:
#' (docs)
#' @examplesShinylive
#' foo <- 1
#' {{ next_example }}
#' bar <- 2
#' @examples
#' (example code)

# using keywords - `{{ prev_example }}`:
#' (docs)
#' bar <- 2
#' @examples
#' (example code)
#' @examplesShinylive
#' foo <- 1
#' {{ prev_example }}

# A typical example would be:
#' (docs)
#' @examplesShinylive
#' library(<package>)
#' interactive <- function() TRUE
#' {{ next_example }}
#' @examples
#' app <- ...
#' if (interactive()) {
#'   shinyApp(app$ui, app$server)
#' }

# multiple apps:
#' (docs)
#' @examplesShinylive
#' @examples
#' (example app 1)
#' @examplesShinylive
#' @examples
#' (example app 2)

# skip parts of example code:
#' (docs)
#' @examples
#' (example code - skipped)
#' @examplesShinylive
#' @examples
#' (example code - included)

# multiple apps with keywords:
#' (docs)
#' @examplesShinylive
#' x <- 1
#' {{ next_example }}
```

```
#' @examples
#' (example app 1)
#' @examplesShinylive
#' y <- 1
#' {{ next_example }}
#' @examples
#' (example app 2)

# combining multiple examples:
#' (docs)
#' @examples
#' (app pre-requisites)
#' @examples
#' (example app)
#' @examplesShinylive
#' {{ paste0(examples, collapse = ", ") }}

# identical to the above example but with a different approach:
#' (docs)
#' @examples
#' (app pre-requisites)
#' @examples
#' (example app)
#' @examplesShinylive
#' {{ paste0(lapply(tags_examples, `[[`, "raw"), collapse = ", ") }}
```

Index