Package: shiny.gems (via r-universe)

March 21, 2025

Type Package

Title Useful Functions and Modules for Shiny Apps

Version 0.0.6

Date 2025-02-21

Author Karsten Weinert

Maintainer Karsten Weinert <karsten.weinert@prognos.com>

Description This package contains several helper functions and demos for Shiny applications. For example, there are examples for exception handling and darkmode support.

License file LICENSE

Depends R (>= 4.1.0)

Imports shiny, bslib, colorspace

Suggests titanic, khroma, vcd, grid, ggplot2

LazyLoad yes

RoxygenNote 7.3.2

Config/pak/sysreqs make zlib1g-dev

Repository https://kweinert.r-universe.dev

RemoteUrl https://github.com/kweinert/shiny.gems

RemoteRef HEAD

RemoteSha 391f6c22d1c1d901a910c1f82bf31afcb957b10f

Contents

`adjust_colors_to_darkmode` *Adjust colors based on lightness*

Description

We may change the colors when entering dark mode. If the color is too dark, we make it a bit lighter. If the color is bright, we make it a bit darker. We use L from the HCL colorspace to determine the lightness/darkness. We use the colorspace::lighten function.

Usage

```
adjust_colors_to_darkmode(colors, threshold = c(30, 70), amount = 0.2)
```

Arguments

`colors`	character vector of colors, e.g. hex codes
`threshold`	numeric, if the L is below the first value, it gets lightened, if above the second value, it gets darkened. It's possible to pass one value only.
`amount`	numeric, how much lighter/darker. Default 0.15

Value

a character of the same length as colors with the (potentially) modified values.

`bs_pal` *Bootstrap Color Palette*

Description

This function uses bslib::bs_current_theme() and bslib::bs_get_variables() to query the root level colors. If dark ist TRUE, it returns the "-dark" variables. The names of the returned vector are however those used for the light mode.

Usage

```
bs_pal(dark = FALSE)
```

Arguments

`dark`	logical, default FALSE

Details

Note that there is a slightly different naming convention for highlight/highlight-bg. In dark mode, these colors are stored under mark-color-dark/mark-bg-dark.

This function should be called in a reactive context.

See https://getbootstrap.com/docs/5.3/customize/color-modes/ for more information on the color mode.

Value

a named character vector, with names body-bg, body-color, body-emphasis-color, body-secondary-color, body-secondary-bg, body-tertiary-color, body-tertiary-bg, headings-color, link-color, link-hover-color, code-color, highlight-color, highlight-bg, border-color, border-color-translucent, form-valid-color, form-valid-border-color, form-invalid-color, form-invalid-border-color

---

colormode_srv *colormode_ui/srv is a shiny module for managing colors; in particular enabling dark mode.*

Description

The server follows the "petite r" approach. It expects a reactiveValues parameter r. It modifies entries of the "colormode

Usage

```
colormode_srv(id = "colormode", r)
```

Arguments

id	character, shiny id. Default "colormode"
r	shiny::reactiveValues object

Details

Currently, it is not possible to save the setting across session. This would require a user management.

The module follows a singleton design pattern, hence the id is preset to "colormode". It is strongly recommended to keep that id.

See colormode_demo to see the module in action, see colormode_srv for implementation details.

Value

the output of shiny::radioButtons()

`colormode_ui` *colormode_ui/srv is a shiny module for managing colors; in particular enabling dark mode.*

Description

The UI produces a subform that can be integrated in a settings/prefences tab. Inspired by the wikipedia mobile version (Feb. 2025), it displays a radiobutton choice between "light", "dark", and "automatic". The default is "light".

Usage

```
colormode_ui(id = "colormode", lang = c("en", "de"), ...)
```

Arguments

`id`	character, shiny id. Default "colormode"
`lang`	character, currently supported are "en" and "de". Default "en".
`...`	further arguments that are passed to shiny::radioButtons(). In particular, "width" and "inline" can be set this way.

Details

For the automatic setting, Javascript is used to determine the local hour. The Javascript code curates a variable "auto_status" that is accessible tin the server module. In particular, the Javascript updates "auto_status" when at 8pm and 6am.

The module follows a singleton design pattern, hence the id is preset to "colormode". It is strongly recommended to keep that id.

See colormode_demo to see the module in action, see colormode_srv for implementation details.

Value

a shiny::div

`exec_safely` *Shiny Version of TryCatch*

Description

Use in reactive context, i.e. inside a server function only.

Usage

```
exec_safely(session, expr)
```

Arguments

`session`	the app session object
`expr`	R expression to evaluate safely

Index