

glasstabs

Glass-morphism Shiny widgets

v0.3.1

Setup

Call once in every UI:

```
library(glasstabs)
ui <- fluidPage(
  useGlassTabs(),
  # widgets go here
)
```

Run a built-in demo:

```
runGlassExample() # list all
runGlassExample("smoke-test")
runGlassExample("basic")
runGlassExample("bs4dash")
```

View changelog in console:

glasstabs_news()

Tab widget — UI

```
glassTabsUI(
  id,           # module id
  ...,         # glassTabPanel()s
  selected = NULL, # initial tab value
  wrap = TRUE,
  compact = FALSE, # reduced spacing
  extra_ui = NULL, # right-side widget
  theme = "dark" # or "light"
)

glassTabPanel(
  value,       # unique tab id
  label,       # button text
  ...,        # pane content
  icon = NULL, # shiny::icon() etc.
  selected = FALSE
)
```

Example:

```
glassTabsUI(
  "main",
  glassTabPanel("a", "Alpha",
    selected = TRUE,
    p("Alpha content")),
  glassTabPanel("b", "Beta",
    icon = icon("star"),
    p("Beta content"))
)
```

Tab widget — server

```
server <- function(input, output, session) {
  # Reactive active tab value
  active <- glassTabsServer("main")
  # With bookmarking (default TRUE)
  active <- glassTabsServer("main",
    bookmark = TRUE)
  observe(print(active()))
}
```

Module pattern:

UI: glassTabsUI(ns("tabs"), ...)

Server: glassTabsServer("tabs")

Pass bare id to server — not ns("tabs").

Tab server actions

```
# Switch active tab
updateGlassTabsUI(session, "main", "b")
```

```
# Show / hide
showGlassTab(session, "main", "admin")
hideGlassTab(session, "main", "admin")
```

```
# Disable / enable (stays visible)
disableGlassTab(session, "main", "b")
enableGlassTab(session, "main", "b")
```

```
# Badge counts (0 or NA = hide)
updateGlassTabBadge(session, "main",
  "inbox", count = 5L)
updateGlassTabBadge(session, "main",
  "inbox", count = 0L)
```

```
# Add / remove at runtime
appendGlassTab(session, "main",
  glassTabPanel("new", "New", p("...")),
  select = TRUE)
removeGlassTab(session, "main", "new")
```

conditionalPanel helper

glassTabCondition() builds the JS condition string:

```
# Instead of:
# "input['main-active_tab'] == 'b'"
conditionalPanel(
  condition =
    glassTabCondition("main", "b"),
  p("Only on tab b")
)
```

```
# Inside a module (same id as UI):
glassTabCondition(ns("tabs"), "b")
```

Dynamic tabs

```
# UI placeholder
glassTabsOutput("dyn_tabs")

# Server render
output$dyn_tabs <- renderGlassTabs({
  panels <- lapply(my_data(),
    function(r)
      glassTabPanel(r$id, r$label,
        p(r$id))
  do.call(glassTabsUI,
    c(list("dyn"), panels))
})
```

JS re-initialises automatically after each render.

Bookmarking

Enable on the app, glassTabsServer() handles save/restore:

Opt out per widget:

```
glassTabsServer("tabs",
  bookmark = FALSE)
```

Tab theming

```
# String preset
glassTabsUI("e", ..., theme = "light")
glassTabsUI("e", ..., theme = "dark")
```

```
# Custom object
glassTabsUI("e", ...,
  theme = glass_tab_theme(
    halo_bg = "rgba(251,191,36,.15)",
    tab_active_text = "#fef3c7"
  )
)
```

)

glass_tab_theme() args:

- tab_text, tab_active_text
- halo_bg, halo_border
- content_bg, content_border
- card_bg, card_text

Multi-select — UI

```
choices <- c(Revenue = "rev",
  Orders = "ord")
```

```
glassMultiSelect(
  inputId,
  choices,
  selected = NULL,
  label = NULL,
  placeholder = "Select...",
  all_label = "All",
  check_style = "checkbox",
  show_style_switcher = FALSE,
  show_select_all = TRUE,
  show_clear_all = TRUE,
  searchable = TRUE,
  theme = "dark"
)
```

Multi-select — server

```
# Reactive: selected() and style()
vals <- glassMultiSelectValue(input, "m")
vals$selected()
vals$style() # "checkbox"|"filled"|"dot"
```

```
# Update
updateGlassMultiSelect(session, "m",
  selected = c("rev", "ord"),
  check_style = "filled")
```

```
# Clear all
updateGlassMultiSelect(session, "m",
  selected = character(0))
```

```
# Filter-tag display (in UI)
glassFilterTags("m")
```

Single-select — UI

```
regions <- c("All" = "all",
  North = "n")
```

```
glassSelect(
  inputId,
  choices,
  selected = NULL,
  label = NULL,
  placeholder = "Pick one...",
  searchable = FALSE,
  clearable = FALSE,
  include_all = FALSE,
  check_style = "checkbox",
  theme = "dark"
)
```

Single-select — server

```
# Reactive value
region <- glassSelectValue(input, "r")
region()
```

```
# Update from server
updateGlassSelect(session, "r",
  selected = "south")
```

```
# Clear
```

```
updateGlassSelect(session, "r",
  selected = character(0))
```

Select theming

```
glassMultiSelect("m", choices,
  theme = glass_select_theme(
    mode = "dark",
    accent_color = "#3b8d80",
    bg_color = "rgba(9,20,42,.97)",
    text_color = "#cfe6ff"
  )
)
```

glass_select_theme() args:

- mode: "dark" or "light"
- bg_color, border_color
- text_color, accent_color
- label_color

Input values

Widget	Input key
tabs	input["id-active_tab"]
multi-select	input\$inputId
multi style	input\$inputId.style
single	input\$inputId

Reactive helpers:

- glassTabsServer() → active()
- glassMultiSelectValue() → vals\$selected()
- glassSelectValue() → region()

bs4Dash integration

```
library(bs4Dash)
ui <- bs4DashPage(
  header = bs4DashNavbar(),
  sidebar = bs4DashSidebar(...),
  body = bs4DashBody(
    useGlassTabs(), # place once here
    bs4TabItems(
      bs4TabItem("explorer",
        bs4Card(width = 12,
          glassTabsUI("tabs",
            glassTabPanel("a", "A", ...),
            glassTabPanel("b", "B", ...))
        )
      )
    )
  )
server <- function(input, output, session) {
  active <- glassTabsServer("tabs")
}
```

Common gotchas

- useGlassTabs() once in UI.
- Modules: ns("tabs") in UI, bare "tabs" in server.
- Duplicate glassTabPanel() values → error.
- selected = tab value, not label.

- | | | | |
|--|---|--|---|
| <ul style="list-style-type: none">• <code>glassSelectValue()</code> returns reactive, not list.• <code>glassMultiSelectValue()</code> returns | <ul style="list-style-type: none">• <code>list(selected(), style())</code>.• Multi-select <code>selected</code> uses values, not labels. | <ul style="list-style-type: none">• After <code>load_all()</code>, reload browser to pick up JS changes. | priglasg.github.io/glasstabs |
|--|---|--|---|