

Package: shinypal (via r-universe)

June 5, 2026

Title Tools to Make Modular and Interactive R Workflow Builders

Version 0.0.0.9000

Depends R (>= 4.1.0)

Imports shiny, shinymeta, DT, bslib (>= 0.11.0), rlang, sortable, htmltools, clipr, purrr, khroma, grDevices, shinycssloaders

Description Contains tools to develop and run modular and interactive shiny apps that use shinymeta to translate GUI-based pipelines into R and RMarkdown code.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libuv1-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-06-05 20:00:20 UTC

RemoteUrl <https://github.com/willgearty/shinypal>

RemoteRef HEAD

RemoteSha ed08a04fe3c0eeba5173de88d9fc041243f815c4

Contents

accordion_panel_remove_button	2
add_shinypal_step	2
clip_observe	3
column_select_observe	4
df_modal_button	4
df_modal_observe	5
df_select_observe	5
file_observe	6
get_chunk	6
get_colors	7

get_int_data	7
get_int_dfs	8
next_step_index	8
select_column_input	9
select_dataset_input	9
set_int_data	10
shinypal_setup	10
shinypal_ui	11
verbatimTextOutput_copy	11
workflow_has_errors	12

Index 13

accordion_panel_remove_button

Accordion panel that includes a remove button

Description

Accordion panel that includes a remove button

Usage

```
accordion_panel_remove_button(ind, ...)
```

Arguments

ind	The index of the step.
...	Additional arguments passed to <code>bslib::accordion_panel()</code> .

add_shinypal_step *Add a step to the report, workflow, and the code chain*

Description

A short description...

Usage

```
add_shinypal_step(
  input,
  ind,
  fun_workflow,
  fun_report,
  code_chain_list,
  libs,
  ec_subs = NULL
)
```

Arguments

input	The input object from the shiny app.
ind	The index of the step to be added.
fun_workflow	A function that generates the UI elements for the workflow.
fun_report	A function that generates the UI elements for the report.
code_chain_list	A list of quoted code bits that will be added to code_chain().
libs	A character vector of R packages required for this step.
ec_subs	An optional list of length 2, where the first element is a metaReactive object and the second element is a callback function. This is used to substitute expansion contexts in the code chain.

clip_observe	<i>Add an observer to a copy button</i>
--------------	---

Description

Generates an event observer that watches for when a copy button is clicked. Upon clicking, the code_expr is evaluated and copied to the clipboard using `clipr::write_clip()`.

Usage

```
clip_observe(input, output, ind, code_expr)
```

Arguments

input	The shiny input object.
output	The shiny output object.
ind	The index of the step.
code_expr	An expression (generated by <code>rlang::expr()</code>) that, when evaluated, returns content to be copied (such as the output from <code>shinymeta::expandChain()</code>).

column_select_observe *Keep a varSelectInput of data.frame column names up-to-date*

Description

Keep a varSelectInput of data.frame column names up-to-date

Usage

```
column_select_observe(input, ind, inputId)
```

Arguments

input	The shiny input object.
ind	The index of the step.
inputId	The id of the <code>shiny::varSelectInput()</code> object.

df_modal_button *Button to show dataframe modal*

Description

Generates a `shiny::actionButton()` that, when clicked, generates a modal for the resulting dataset for the specified step. Make sure to set up a corresponding observer using `df_modal_observe()`.

Usage

```
df_modal_button(ind, text = "View data")
```

Arguments

ind	The index of the step.
text	The text to display on the button.

df_modal_observe	<i>Show a modal with a reactable data.frame</i>
------------------	---

Description

Show a modal with a reactable data.frame

Usage

```
df_modal_observe(input, output, ind, df_name)
```

Arguments

input	The shiny input object.
output	The shiny output object.
ind	The index of the step.
df_name	The name of the data.frame to be displayed.

df_select_observe	<i>Keep a selectInput of intermediate data.frames up-to-date</i>
-------------------	--

Description

Keep a selectInput of intermediate data.frames up-to-date

Usage

```
df_select_observe(input, ind)
```

Arguments

input	The shiny input object.
ind	The index of the step.

file_observe	<i>Observe a file input to be included in the download bundle</i>
--------------	---

Description

Observe a file input to be included in the download bundle

Usage

```
file_observe(input, inputId)
```

Arguments

input	The shiny input object.
inputId	The id of the <code>shiny::fileInput()</code> object. of the file in the code.

get_chunk	<i>Get the expanded code chunk for a registered step</i>
-----------	--

Description

Returns the code chunk for ind, expanded with a shared context across all currently-registered steps. Use this inside a reactive consumer (e.g., `renderPrint()`, `observeEvent()`) to display or copy a step's code. The chunk is rebuilt whenever any dependency changes.

Usage

```
get_chunk(ind)
```

Arguments

ind	The index of the step.
-----	------------------------

Value

A code object suitable for printing or passing to `shiny::displayCodeModal()`, or NULL if the step is not registered.

get_colors	<i>Generate a background color and font color for a step based on its index</i>
------------	---

Description

Generate a background color and font color for a step based on its index

Usage

```
get_colors(ind)
```

Arguments

ind	The index of the step.
-----	------------------------

Value

A list containing the color and background CSS styles for the step.

get_int_data	<i>Get an intermediate data object</i>
--------------	--

Description

Get an intermediate data object

Usage

```
get_int_data(name)
```

Arguments

name	The name of the data object to retrieve.
------	--

get_int_dfs	<i>Get the names of all intermediate data.frames for a given step</i>
-------------	---

Description

Get the names of all intermediate data.frames for a given step

Usage

```
get_int_dfs(ind)
```

Arguments

ind	The index of the step.
-----	------------------------

next_step_index	<i>Get the next workflow step index</i>
-----------------	---

Description

Returns a unique, monotonically increasing index for a new workflow step. Each call increments a server-side counter. The counter is initialised by `shinyPAL_setup()` and is deliberately never reset while the session is running.

Usage

```
next_step_index()
```

Value

A single positive integer to use as the new step's index.

select_column_input *Select input to choose a column from a specified dataset*

Description

This should be paired with [select_dataset_input\(\)](#).

Usage

```
select_column_input(ind, label = "Choose a column:", default = NULL, ...)
```

Arguments

ind	The index of the step.
label	The label for the select input.
default	The default value for the select input.
...	Additional arguments passed to shiny::varSelectInput() .

select_dataset_input *Select input to choose a shinydal intermediate dataset*

Description

Select input to choose a shinydal intermediate dataset

Usage

```
select_dataset_input(ind, label = "Choose a dataset:")
```

Arguments

ind	The index of the step.
label	The label for the select input.

set_int_data	<i>Set an intermediate data object</i>
--------------	--

Description

Set an intermediate data object

Usage

```
set_int_data(obj, name)
```

Arguments

obj	A reactive data object to store.
name	A name to store data object as.

shinyPAL_setup	<i>Setup shinyPAL</i>
----------------	-----------------------

Description

A function to set up the shinyPAL environment. This function should be called at the top of the server function of your shiny app.

Usage

```
shinyPAL_setup(
  input,
  output,
  session,
  modules,
  download_filename = "shinyPAL_script.zip",
  download_template = "./modules/test_report.qmd"
)
```

Arguments

input	The shiny input object.
output	The shiny output object.
session	The shiny session object.
modules	A character vector of paths to independent modules.
download_filename	The name of the file to download when the user clicks the download button.
download_template	The path to the Quarto markdown template file that will be used to generate the report.

Details

Each module should have a `ui-aux.R` file that defines the UI elements for that module in the "Workflow" accordion and a `server.R` file that defines the server-side logic for the module.

shinyPal_ui	<i>UI for ShinyPal</i>
-------------	------------------------

Description

A function to create the UI for the ShinyPal app. This function should be called in the UI function of your shiny app.

Usage

```
shinyPal_ui(modules)
```

Arguments

`modules` A character vector of paths to independent modules.

Details

Each module should have a `ui-main.R` file that defines the UI elements for that module in the "Steps" accordion.

verbatimTextOutput_copy	<i>Text element with a button to copy the code to the clipboard</i>
-------------------------	---

Description

Render a reactive output variable as text within an application page. Uses `shiny::verbatimTextOutput()` which is usually paired with `shiny::renderPrint()` and provides fixed-width text in a `<pre>`. Make sure to set up a corresponding observer using `clip_observe()`.

Usage

```
verbatimTextOutput_copy(ind)
```

Arguments

`ind` The index of the step.

workflow_has_errors *Check whether the workflow has incomplete or errored steps*

Description

Returns TRUE if any step in the current workflow cannot be expanded into the downloadable script due to failing req()/validate() checks.

Usage

```
workflow_has_errors()
```

Value

A length-one logical.

Index

accordion_panel_remove_button, 2
add_shiny_pal_step, 2

bslib::accordion_panel(), 2

clip_observe, 3
clip_observe(), 11
clipr::write_clip(), 3
column_select_observe, 4

df_modal_button, 4
df_modal_observe, 5
df_modal_observe(), 4
df_select_observe, 5

file_observe, 6

get_chunk, 6
get_colors, 7
get_int_data, 7
get_int_dfs, 8

next_step_index, 8

rlang::expr(), 3

select_column_input, 9
select_dataset_input, 9
select_dataset_input(), 9
set_int_data, 10
shiny::actionButton(), 4
shiny::fileInput(), 6
shiny::renderPrint(), 11
shiny::varSelectInput(), 4, 9
shiny::verbatimTextOutput(), 11
shiny::displayCodeModal(), 6
shiny::expandChain(), 3
shiny::setup, 10
shiny::setup(), 8
shiny::ui, 11

verbatimTextOutput_copy, 11

workflow_has_errors, 12