Package: r6methods (via r-universe)

October 25, 2024

Type Package

Title Make Methods for R6 Classes

Version 0.1.1

Description Generate boilerplate code for R6 classes. Given R6 class create getters and/or setters for selected class fields or use RStudio addins to insert methods straight into class definition.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports glue, rstudioapi, miniUI, shiny, dplyr, magrittr, stringr, purrr

Suggests mockery, R6, testthat (>= 3.0.0)

Config/testthat/edition 3

URL `https://github.com/jakubsob/r6methods`

BugReports `https://github.com/jakubsob/r6methods/issues`

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

RemoteUrl https://github.com/jakubsob/r6methods

RemoteRef HEAD

RemoteSha 4161d6fc070e002a0027261f4ae18ccd3ea31144

Contents

`extract_class` *Extract R6 Class*

Description

Extract R6 Class

Usage

```
extract_class(content, start_pos = 1)
```

Arguments

`content`	Character, content of a file or a string
`start_pos`	Integer, row position of cursor. Serves as starting point to search for class definition

Value

A list with fields

- class_contentCharacter, extracted class definition
- startInteger, start position of class definition within ‘content‘
- endInteger, end position of class definition within ‘content‘

`find_closing` *Find Closing*

Description

Find position of closing character to first encountered opening character

Usage

```
find_closing(text, opening = "\\(", closing = "\\)")
```

Arguments

`text`	Character, text to search
`opening`	Opening character
`closing`	Closing character

Value

Integer, position of closing character

`get_cursor_pos` *Get Cursor Position From Active Document*

Description

Get Cursor Position From Active Document

Usage

```
get_cursor_pos(context)
```

Arguments

`context`	Active document context

Value

Integer, position of cursor in text

`insert_methods` *Insert Methods*

Description

Insert Methods

Usage

```
insert_methods(
  content,
  start_pos = 1,
  field = c("all", "public", "private"),
  method = c("both", "get", "set"),
  add_roxygen = TRUE
)
```

Arguments

`content`	Character, content of the file or a string
`start_pos`	Integer, position of cursor within ‘content‘. Number of characters before the cursor.
`field`	Character, fields for which to create method. May be "all", "public", "private" or name of class field. Multiple values allowed.
`method`	Character, methods to create. One of "both", "get", "set"
`add_roxygen`	Logical, whether to add roxygen description of method

Value

Character, modified `content` with injected methods

`insert_methods_addin` *An addin for inserting methods straigth into the source file*

Description

An addin for inserting methods straigth into the source file

Usage

```
insert_methods_addin()
```

Value

No return value, called for side effects

`insert_methods_addin_gadget` *Insert methods addin gadget*

Description

Insert methods addin gadget

Usage

```
insert_methods_addin_gadget()
```

Value

No return value, called for side effects

`make_gadget` *Make Gadget*

Description

Create gadget for generating R6 methods. Action after clicking ‘Done‘ button is defined by ‘done_fun‘.

Usage

```
make_gadget(title, title_bar, done_fun)
```

Arguments

`title`	Character, title of gadget window
`title_bar`	Character, gadget title bar
`done_fun`	Function to be used after clicking ‘Done‘ button

Value

Function creating and running a Shiny gadget

`make_methods` *Make methods*

Description

Make methods

Usage

```
make_methods(
  r6,
  field = c("all", "public", "private", names(r6$public_fields),
    names(r6$private_fields)),
  method = c("both", "get", "set"),
  add_roxygen = TRUE
)
```

Arguments

`r6`	R6 class for which to create methods
`field`	Character, fields for which to create method. May be "all", "public", "private" or name of class field. Multiple values allowed.
`method`	Character, methods to create. One of "both", "get", "set"
`add_roxygen`	Logical, whether to add roxygen description of method

Value

Character containing generated methods to put into class definition

Examples

```
Example <- R6::R6Class("Example", list(public_field = NULL), list(private_field = NULL))
make_methods(Example)
make_methods(Example, "private", "get")
make_methods(Example, "private_field", c("get", "set"))
make_methods(Example, "public_field", c("both"))
```

`make_methods_addin` *Make Methods Addin*

Description

Make Methods Addin

Usage

```
make_methods_addin()
```

Value

No return value, called for side effects

`make_methods_addin_gadget` *Make methods addin gadget*

Description

Make methods addin gadget

Usage

```
make_methods_addin_gadget()
```

Value

No return value, called for side effects

`make_method_str` *Make method string*

Description

Make method string

Usage

```
make_getter_method_str(field, is_public = TRUE, add_roxygen = TRUE)

make_setter_method_str(field, is_public = TRUE, add_roxygen = TRUE)
```

Arguments

`field`	Character name of class field
`is_public`	Logical, whether the field is in public list
`add_roxygen`	Logical, whether to add roxygen description of method

Value

Character containing method definition

`ReactiveR6` *ReactiveR6*

Description

This class allows you to make your R6 class reactive by inheriting from `ReactiveR6` By calling `private$invalidate()` in a method you can invalidate the class in a controlled way, i.e. only when specific methods are called. See example how to use it.

Inspired by `https://community.rstudio.com/t/good-way-to-create-a-reactive-aware-r6-class/84890`

Methods

Public methods:

- `ReactiveR6$reactive()`
- `ReactiveR6$clone()`

Method `reactive()`**:** Call this method to make object instance reactive

Usage:

```
ReactiveR6$reactive()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
ReactiveR6$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## Not run:
if (interactive()) {
library(shiny)
library(r6methods)

Counter <-  R6::R6Class(
  "Counter",
  inherit = ReactiveR6,
  public = list(
    increment = function() {
      private$counter <- private$counter + 1
      private$invalidate()
    },
    decrement = function() {
      private$counter <- private$counter - 1
      private$invalidate()
    },
    silent_increment = function() {
      private$counter <- private$counter + 1
    },
    get_counter = function() {
      private$counter
    }
  ),
  private = list(
    counter = 0
  )
)

counter <- Counter$new()$reactive()

shinyApp(
  fluidPage(
    actionButton("increment", "Increment"),
    actionButton("decrement", "Decrement"),
    actionButton("silent_increment", "Silent increment"),
    textOutput("value")
  ),
  function(input, output, session) {
    observeEvent(input$increment, {
      counter()$increment()
    })
    observeEvent(input$decrement, {
      counter()$decrement()
    })

    observeEvent(input$silent_increment, {
      counter()$silent_increment()
    })

    output$value <- renderText({
      counter()$get_counter()
    })
  }
)
}

## End(Not run)
```

source_class *Source class*

Description

Sources R6 class from text, prepends namespace to ‘R6Class‘ in order to not require ‘R6‘ to be loaded.

Usage

```
source_class(txt)
```

Arguments

`txt`	Character, text containing class definition

Value

R6 class

Index