Package 'findInFiles'

August 3, 2021

Type Package

Title Find Pattern in Files

Version 0.3.0

Description Creates a HTML widget which displays the results of searching for a pattern in files in a given folder. The results can be viewed in the 'RStudio' viewer pane, included in a 'R Markdown' document or in a 'Shiny' app.

License GPL-3

Encoding UTF-8

SystemRequirements grep

Imports htmlwidgets, stringr, crayon, vctrs, tibble, stringi

Suggests shiny

URL https://github.com/stla/findInFiles

BugReports https://github.com/stla/findInFiles/issues

RoxygenNote 7.1.1

NeedsCompilation no

Author Stéphane Laurent [aut, cre],
Rob Burns [ctb, cph] ('ansi-to-html' library)

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2021-08-03 09:10:02 UTC

R topics documented:

FIF2dataframe *Output of 'findInFiles' as a dataframe*

Description

Returns the results of `findInFiles` in a dataframe, when the option `output = "viewer+tibble"` or `output = "tibble"` is used.

Usage

```
FIF2dataframe(fif)
```

Arguments

`fif`	the output of `findInFiles` used with the option `output = "viewer+tibble"` or `output = "tibble"`

Value

The results of `findInFiles` in a dataframe.

Examples

```
folder <- system.file("example", package = "findInFiles")
fif <- findInFiles("R", "function", root = folder, output = "viewer+tibble")
FIF2dataframe(fif)
fif
```

FIF2tibble *Output of 'findInFiles' as a tibble*

Description

Returns the results of `findInFiles` in a tibble, when the option `output = "viewer+tibble"` is used.

Usage

```
FIF2tibble(fif)
```

Arguments

`fif`	the output of `findInFiles` used with the option `output = "viewer+tibble"`

Value

The results of `findInFiles` in a tibble.

Examples

```
folder <- system.file("example", package = "findInFiles")
fif <- findInFiles("R", "function", root = folder, output = "viewer+tibble")
FIF2tibble(fif)
fif
```

`findInFiles` *Find pattern in files*

Description

Find a pattern in some files.

Usage

```
findInFiles(
  ext,
  pattern,
  depth = NULL,
  wholeWord = FALSE,
  ignoreCase = FALSE,
  perl = FALSE,
  excludePattern = NULL,
  excludeFoldersPattern = NULL,
  root = ".",
  output = "viewer"
)
```

Arguments

`ext`	file extension, e.g. `"R"` or `"js"`
`pattern`	pattern to search for, a regular expression, e.g. `"function"` or `"^function"`
`depth`	depth of the search, `NULL` or a negative number for an entire recursive search (subdirectories, subdirectories of subdirectories, etc.), otherwise a positive integer: `0` to search in the root directory only, `1` to search in the root directory and its subdirectories, etc.
`wholeWord`	logical, whether to match the whole pattern
`ignoreCase`	logical, whether to ignore the case
`perl`	logical, whether `pattern` is a Perl regular expression
`excludePattern`	a pattern; exclude from search the files and folders which match this pattern
`excludeFoldersPattern`	a pattern; exclude from search the folders which match this pattern
`root`	path to the root directory to search from
`output`	one of `"viewer"`, `"tibble"` or `"viewer+tibble"`; see examples

Value

A tibble if `output="tibble"`, otherwise a `htmlwidget` object.

Examples

```
library(findInFiles)
folder <- system.file("example", package = "findInFiles")
findInFiles("R", "function", root = folder)

findInFiles("R", "function", root = folder, output = "tibble")

fif <- findInFiles("R", "function", root = folder, output = "viewer+tibble")
FIF2tibble(fif)
FIF2dataframe(fif)
fif

folder <- system.file("www", "shared", package = "shiny")
findInFiles("css", "outline", excludePattern = "*.min.css", root = folder)
```

---

findInFiles-shiny *Shiny bindings for* findInFiles

Description

Output and render functions for using `findInFiles` within Shiny applications and interactive Rmd documents.

Usage

```
FIFOutput(outputId, width = "100%", height = "400px")

renderFIF(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	output variable to read from
`width, height`	a valid CSS unit (like `"100%"`, `"400px"`, `"auto"`) or a number, which will be coerced to a string and have `"px"` appended
`expr`	an expression that generates a '`findInFiles`' widget
`env`	the environment in which to evaluate `expr`
`quoted`	logical, whether `expr` is a quoted expression (with `quote()`)

Value

`FIFOutput` returns an output element that can be included in a Shiny UI definition, and `renderFIF` returns a `shiny.render.function` object that can be included in a Shiny server definition.

Examples

```
library(findInFiles)
library(shiny)

onKeyDown <- HTML(
  'function onKeyDown(event) {',
  '  var key = event.which || event.keyCode;',
  '  if(key === 13) {',
  '    Shiny.setInputValue(',
  '      "pattern", event.target.value, {priority: "event"}',
  '    );',
  '  }',
  '}'
)

ui <- fluidPage(
  tags$head(tags$script(onKeyDown)),
  br(),
  sidebarLayout(
    sidebarPanel(
      selectInput(
        "ext", "Extension",
        choices = c("R", "js", "css")
      ),
      tags$div(
        class = "form-group shiny-input-container",
        tags$label(
          class = "control-label",
          "Pattern"
        ),
        tags$input(
          type = "text",
          class = "form-control",
          onkeydown = "onKeyDown(event);",
          placeholder = "Press Enter when ready"
        )
      ),
      numericInput(
        "depth", "Depth (set -1 for unlimited depth)",
        value = 0, min = -1, step = 1
      ),
      checkboxInput(
        "wholeWord", "Whole word"
      ),
      checkboxInput(
        "ignoreCase", "Ignore case"
      )
    ),
    mainPanel(
      FIFOutput("results")
    )
  )
```

```
)


server <- function(input, output){

  output[["results"]] <- renderFIF({
    req(input[["pattern"]])
    findInFiles(
      ext = isolate(input[["ext"]]),
      pattern = input[["pattern"]],
      depth = isolate(input[["depth"]]),
      wholeWord = isolate(input[["wholeWord"]]),
      ignoreCase = isolate(input[["ignoreCase"]])
    )
  })

}

if(interactive()){
  shinyApp(ui, server)
}
```

Index