Package ‘supreme’

July 8, 2020

Title Modeling Tool for 'Shiny' Applications Developed with Modules

Version 1.1.0

Description A modeling tool helping users better structure 'Shiny' applications developed with 'Shiny' modules. Users are able to: 1. Visualize relationship of modules in existing applications 2. Design new applications from scratch.

License MIT + file LICENSE

URL `https://strboul.github.io/supreme/`

BugReports `https://github.com/strboul/supreme/issues`

Depends R (>= 3.6.0)

Imports stats, utils, yaml (>= 2.2.0), nomnoml, shiny (>= 1.5.0)

Suggests testthat (>= 2.1.0), covr, knitr, rmarkdown, digest (>= 0.6.23)

Language en-US

LazyData true

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Metin Yazici [aut, cre, cph]

Maintainer Metin Yazici <`stradivariusboul+r@gmail.com`>

Repository CRAN

Date/Publication 2020-07-08 08:30:02 UTC

R topics documented:

as.data.frame.supreme 2
example_app_path 2
example_yaml 3
graph 4

src_file 5
src_yaml 6
supreme 7

Index **8**

as.data.frame.supreme *Turn supreme data into a* data.frame

Description

Turn supreme data into a data.frame

Usage

```
## S3 method for class 'supreme'
as.data.frame(x, ...)
```

Arguments

x	a supreme object.
...	methods to be passed onto.

Value

a data.frame.

Examples

```
paths <- example_app_path()
sp <- supreme(src_file(paths))
as.data.frame(sp)
```

example_app_path *Get paths to supreme example*

Description

The example Shiny application to demonstrate all the capabilities of what supreme offers.

Usage

```
example_app_path(file = NULL)
```

Arguments

file	file names. If no file names are put (which path is NULL), then all the example file paths will be listed.

Value

a character vector containing the R file path for the example.

See Also

Other source examples: `example_yaml()`

Examples

```
files <- example_app_path(c("app", "module-customers"))
supreme(src_file(files))
```

`example_yaml` *Get YAML to supreme example*

Description

Get YAML to supreme example

Usage

```
example_yaml()
```

Value

a character vector containing the YAML file path for the example.

See Also

Other source examples: `example_app_path()`

Examples

```
yaml <- example_yaml()
supreme(src_yaml(yaml))
```

graph *Make a UML graph of Shiny modules*

Description

Creates a *UML-like* graph from your 'Shiny application' developed with modules.

Usage

```
graph(x, fields = NULL, styles = NULL, options = NULL)
```

Arguments

`x` a `supreme` object.

`fields` optional. name of the fields to include in the graph. The possible values can be found at `getOption("SUPREME_MODEL_REQUIRED_FIELDS")` and `getOption("SUPREME_MODEL_OPTION` By default, the required fields such as the "name" field always visible. There are no ways to exclude the required fields. This parameter is set to `NULL` as default.

`styles` optional. a named list to apply custom styles on the graph nodes. A full list of the available styles can be seen from: nomnoml: Custom classifier styles

`options` optional. custom options for the whole graph. A full list of the available options can be seen from: nomnoml: Directives

Details

The graph call uses the `nomnoml` tool to draw a UML diagram of the Shiny application.

Value

a `supreme` graph.

References

nomnoml: The sassy UML diagram renderer

Examples

```
# create a graph:
path <- example_yaml()
sp <- supreme(src_yaml(path))
graph(sp)

# filter fields, only return the certain fields in the graph entities:
graph(sp, fields = c("input", "return"))

# style entites:
graph(sp, styles = list(
 "server" = list(fill = "#ff0", "underline", "bold"),
 "module_modal_dialog" = list(fill = "lightblue", "dashed", visual = "note")
))

# style entities having a word "tab" in it:
sp_df <- as.data.frame(sp) # turn supreme object to data.frame
tab_modules <- sp_df$name[grep("_tab_", sp_df$name)]
styles <- lapply(seq_along(tab_modules), function(x) list(fill = "orange"))
names(styles) <- tab_modules
graph(sp, styles = styles)

# set graph options:
graph(sp, options = list(
  direction = "right",
  fontSize = 10,
  title = "Model application"
))
```

src_file *Read R files*

Description

Read files contain at least one Shiny application.

Usage

```
src_file(x)
```

Arguments

x	a file path.

Value

A src_file object.

See Also

Other source functions: src_yaml()

Examples

```
paths <- example_app_path()
s <- supreme(src_file(paths))
```

src_yaml *Read a YAML file containing a model*

Description

Reads an object or a file in YAML format and returns a model YAML object.

Usage

```
src_yaml(file = NULL, text = NULL)
```

Arguments

file	file path to a YAML file.
text	a YAML formatted character string.

Value

A `src_yaml` object.

See Also

Other source functions: `src_file()`

Examples

```
## Read from a file:
path <- example_yaml()
src_yaml(path)

## Read from an (text) object:
model <- "
- name: childModuleA
  input: [input.data, reactive]
  src: package

- name: childModuleB
  input: selected.model
"
src_yaml(text = model)
```

`supreme` *Create a supreme object*

Description

Create a supreme object

Usage

```
supreme(x)
```

Arguments

`x`	a valid source input.

Value

a `supreme` object.

Examples

```
path <- example_app_path()
supreme(src_file(path))
```

Index

∗ **source examples**
example_app_path, 2
example_yaml, 3
∗ **source functions**
src_file, 5
src_yaml, 6

as.data.frame.supreme, 2

example_app_path, 2, *3*
example_yaml, *3*, 3

graph, 4

src_file, 5, *6*
src_yaml, *5*, 6
supreme, 7