

Package ‘makepipe’

November 23, 2021

Title Pipeline Tools Inspired by 'GNU Make'

Version 0.1.0

Description A suite of tools for transforming an existing workflow into a self-documenting pipeline with very minimal upfront costs. Segments of the pipeline are specified in much the same way a 'Make' rule is, by declaring an executable recipe (which might be an R script), along with the corresponding targets and dependencies. When the entire pipeline is run through, only those recipes that need to be executed will be. Meanwhile, execution metadata is captured behind the scenes for later inspection.

License GPL (>= 3)

URL <https://kinto-b.github.io/makepipe/>,
<https://github.com/kinto-b/makepipe>

BugReports <https://github.com/kinto-b/makepipe/issues>

Imports cli, R6, utils, visNetwork

Suggests knitr, covr, testthat (>= 3.0.0), withr, rmarkdown

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.1.2

VignetteBuilder knitr

NeedsCompilation no

Author Kinto Behr [aut, cre, cph]

Maintainer Kinto Behr <kinto.behr@gmail.com>

Repository CRAN

Date/Publication 2021-11-23 05:30:02 UTC

R topics documented:

make_register	2
make_with_recipe	3
make_with_source	5
out_of_date	6
pipeline-accessors	7
pipeline-vis	8

Index	10
--------------	-----------

make_register	<i>Register objects to be returned from make_with_source</i>
---------------	--

Description

It is sometimes useful to have access to certain objects which are generated as side-products in a source script which yields as a main-product one or more targets. Typically these objects are used for checking that the targets were produced as expected.

Usage

```
make_register(value, name, quiet = FALSE)
```

Arguments

value	A value to be registered in a source script and returned as part of the Segment
name	A variable name, given as a character string. No coercion is done, and the first element of a character vector of length greater than one will be used, with a warning.
quiet	A logical determining whether or not warnings are signaled when <code>make_register()</code> is called outside of a 'makepipe' pipeline

Value

value invisibly

Examples

```
## Not run:
# Imagine this is part of your source script:
x <- readRDS("input.Rds")
x <- do_stuff(x)
chk <- do_check(x)
make_register(chk, "x_check")
saveRDS(x, "output.Rds")

# You will have access to `chk` in your pipeline script:
```

```

step_one <- make_with_source(
  "source.R",
  "output.Rds",
  "input.Rds",
)
step_one$result$chk

## End(Not run)

```

make_with_recipe *Make targets out of dependencies using a recipe*

Description

Make targets out of dependencies using a recipe

Usage

```

make_with_recipe(
  recipe,
  targets,
  dependencies = NULL,
  packages = NULL,
  envir = new.env(parent = parent.frame()),
  quiet = getOption("makepipe.quiet"),
  force = FALSE,
  ...
)

```

Arguments

recipe	A chunk of R code which makes the targets
targets	A character vector of paths to files
dependencies	A character vector of paths to files which the targets depend on
packages	A character vector of names of packages which targets depend on
envir	The environment in which to execute the source or recipe. By default, execution will take place in a fresh environment whose parent is the calling environment.
quiet	A logical determining whether or not messages are signaled
force	A logical determining whether or not execution of the source or recipe will be forced (i.e. happen whether or not the targets are out-of-date)
...	Additional parameters to pass to <code>base::eval()</code>

Value

A Segment object containing execution metadata.

See Also

Other make: [make_with_source\(\)](#)

Examples

```
## Not run:
# Merge files in fresh environment if raw data has been updated since last
# merged
make_with_recipe(
  recipe = {
    dat <- readRDS("data/raw_data.Rds")
    pop <- readRDS("data/pop_data.Rds")
    merged_dat <- merge(dat, pop, by = "id")
    saveRDS(merged_dat, "data/merged_data.Rds")
  },
  targets = "data/merged_data.Rds",
  dependencies = c("data/raw_data.Rds", "data/raw_pop.Rds")
)

# Merge files in current environment if raw data has been updated since last
# merged. (If recipe executed, all objects bound in source will be available
# in current env).
make_with_recipe(
  recipe = {
    dat <- readRDS("data/raw_data.Rds")
    pop <- readRDS("data/pop_data.Rds")
    merged_dat <- merge(dat, pop, by = "id")
    saveRDS(merged_dat, "data/merged_data.Rds")
  },
  targets = "data/merged_data.Rds",
  dependencies = c("data/raw_data.Rds", "data/raw_pop.Rds"),
  envir = environment()
)

# Merge files in global environment if raw data has been updated since last
# merged. (If source executed, all objects bound in source will be available
# in global env).
make_with_recipe(
  recipe = {
    dat <- readRDS("data/raw_data.Rds")
    pop <- readRDS("data/pop_data.Rds")
    merged_dat <- merge(dat, pop, by = "id")
    saveRDS(merged_dat, "data/merged_data.Rds")
  },
  targets = "data/merged_data.Rds",
  dependencies = c("data/raw_data.Rds", "data/raw_pop.Rds"),
  envir = globalenv()
)

## End(Not run)
```

make_with_source *Make targets out of dependencies using a source file*

Description

Make targets out of dependencies using a source file

Usage

```
make_with_source(  
  source,  
  targets,  
  dependencies = NULL,  
  packages = NULL,  
  envir = new.env(parent = parent.frame()),  
  quiet = getOption("makepipe.quiet"),  
  force = FALSE,  
  ...  
)
```

Arguments

source	The path to an R script which makes the targets
targets	A character vector of paths to files
dependencies	A character vector of paths to files which the targets depend on
packages	A character vector of names of packages which targets depend on
envir	The environment in which to execute the source or recipe. By default, execution will take place in a fresh environment whose parent is the calling environment.
quiet	A logical determining whether or not messages are signaled
force	A logical determining whether or not execution of the source or recipe will be forced (i.e. happen whether or not the targets are out-of-date)
...	Additional parameters to pass to <code>base::source()</code>

Value

A Segment object containing execution metadata.

See Also

Other make: [make_with_recipe\(\)](#)

Examples

```
## Not run:
# Merge files in fresh environment if raw data has been updated since last
# merged
make_with_source(
  source = "merge_data.R",
  targets = "data/merged_data.Rds",
  dependencies = c("data/raw_data.Rds", "data/raw_pop.Rds")
)

# Merge files in current environment if raw data has been updated since last
# merged. (If source executed, all objects bound in source will be available
# in current env).
make_with_source(
  source = "merge_data.R",
  targets = "data/merged_data.Rds",
  dependencies = c("data/raw_data.Rds", "data/raw_pop.Rds"),
  envir = environment()
)

# Merge files in global environment if raw data has been updated since last
# merged. (If source executed, all objects bound in source will be available
# in global env).
make_with_source(
  source = "merge_data.R",
  targets = "data/merged_data.Rds",
  dependencies = c("data/raw_data.Rds", "data/raw_pop.Rds"),
  envir = globalenv()
)

## End(Not run)
```

out_of_date

Check if targets are out-of-date vis-a-vis their dependencies

Description

Check if targets are out-of-date vis-a-vis their dependencies

Usage

```
out_of_date(targets, dependencies, packages = NULL)
```

Arguments

targets A character vector of paths to files

dependencies A character vector of paths to files which the targets depend on
 packages A character vector of names of packages which targets depend on

Value

TRUE if any of targets are older than any of dependencies or if any of targets do not exist;
 FALSE otherwise

Examples

```
## Not run:
out_of_date("data/processed_data.Rds", "data/raw_data.Rds")

## End(Not run)
```

pipeline-accessors *Access and interface with Pipeline.*

Description

`get_pipeline()`, `set_pipeline()` access and modify the current *active* pipeline, while all other helper functions do not affect the active pipeline

Usage

```
is_pipeline(pipeline)

set_pipeline(pipeline)

get_pipeline()
```

Arguments

pipeline A pipeline. See [Pipeline](#) for more details.

See Also

Other pipeline: [Pipeline](#), [pipeline-vis](#)

Examples

```
## Not run:
# Build up a pipeline from scratch and save it out
set_pipeline(Pipeline$new())
# A series of `make_with_*()` blocks go here...
saveRDS(get_pipeline(), "data/my_pipeline.Rds")

## End(Not run)
```

 pipeline-vis

Visualise the Pipeline.

Description

Produce an HTML flowchart visualisation of the pipeline.

Usage

```
show_pipeline(pipeline = get_pipeline(), tooltips = NULL, labels = NULL, ...)
```

```
save_pipeline(
  file,
  pipeline = get_pipeline(),
  tooltips = NULL,
  labels = NULL,
  selfcontained = TRUE,
  background = "white",
  ...
)
```

Arguments

pipeline	A pipeline. See Pipeline for more details.
tooltips	A named character vector mapping nodes in the pipeline onto tooltips to display on hover-over.
labels	A named character vector mapping nodes in the pipeline onto labels to display beside them.
...	Arguments (other than nodes and edges) to pass to <code>visNetwork::visNetwork()</code>
file	File to save HTML into
selfcontained	Whether to save the HTML as a single self-contained file (with external resources base64 encoded) or a file with external resources placed in an adjacent directory.
background	Text string giving the html background color of the widget. Defaults to white.

Details

Tooltips and labels must be supplied as named character vector where the names correspond to the filepaths of nodes (i.e. targets, dependencies, or source scripts)

See Also

Other pipeline: [Pipeline](#), [pipeline-accessors](#)

Examples

```
## Not run:
# Run pipeline
make_with_source(
  "recode.R",
  "data/0 raw_data.R",
  "data/1 data.R"
)
make_with_source(
  "merge.R",
  c("data/1 data.R", "data/0 raw_pop.R"),
  "data/2 data.R"
)

# Visualise pipeline with custom tooltips
show_pipeline(tooltips = c(
  "data/0 raw_data.R" = "Raw survey data",
  "data/0 raw_pop.R" = "Raw population data",
  "data/1 data.R" = "Survey data with recodes applied",
  "data/2 data.R" = "Survey data with demographic variables merged in"
))

## End(Not run)
```

Index

* **make**

make_with_recipe, 3

make_with_source, 5

* **pipeline**

pipeline-accessors, 7

pipeline-vis, 8

get_pipeline (pipeline-accessors), 7

is_pipeline (pipeline-accessors), 7

make_register, 2

make_with_recipe, 3, 5

make_with_source, 4, 5

out_of_date, 6

Pipeline, 7, 8

pipeline-accessors, 7

pipeline-vis, 8

save_pipeline (pipeline-vis), 8

set_pipeline (pipeline-accessors), 7

show_pipeline (pipeline-vis), 8