Package: bookdowntargets (via r-universe)

October 16, 2024

Title Targets for bookdown pipeline

Description Provides a targets pipeline extension for bookdown
projects. For the underlying methodology, please refer to the
documentation of 'targets' <doi:10.21105/joss.02959>.

URL https://github.com/mps9506/bookdowntargets

BugReports https://github.com/mps9506/bookdowntargets/issues

Version 0.1.2

Maintainer Michael Schramm <mpschramm@gmail.com>

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Imports bookdown, fs (>= 1.4.2), targets, tarchetypes, withr (>=
2.1.2)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Suggests rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://mps9506.r-universe.dev

RemoteUrl https://github.com/mps9506/bookdowntargets

RemoteRef HEAD

RemoteSha e2ac8c8073bbe5f6199b8555ccf3a39386b37975

Contents

1

---

bookdowntargets-package

bookdowntargets: tarchetype pipeline for bookdown reports.

Description

A pipeline toolkit for R, the `targets` package brings together function-oriented programming and Make-like declarative pipelines for Statistics and data science. The bookdowntargets package provides a convenient helper function to create specialized targets.

---

tar_render_book   *Target with Bookdown*

Description

Shorthand to render a book via [bookdown::render_book](#) as part of a `targets` pipeline.

Usage

```
tar_render_book(
  name,
  path,
  tidy_eval = targets::tar_option_get("tidy_eval"),
  packages = targets::tar_option_get("packages"),
  library = targets::tar_option_get("library"),
  error = targets::tar_option_get("error"),
  deployment = "main",
  priority = targets::tar_option_get("priority"),
  resources = targets::tar_option_get("resources"),
  retrieval = targets::tar_option_get("retrieval"),
  cue = targets::tar_option_get("cue"),
  description = targets::tar_option_get("description"),
  ...
)
```

Arguments

name   Symbol, name of the target. A target name must be a valid name for a symbol in
                R, and it must not start with a dot. Subsequent targets can refer to this name sym-
                bolically to induce a dependency relationship: e.g. `tar_target(downstream_target,`
                `f(upstream_target))` is a target named `downstream_target` which depends
                on a target `upstream_target` and a function `f()`. In addition, a target's name
                determines its random number generator seed. In this way, each target runs with
                a reproducible seed so someone else running the same pipeline should get the
                same results, and no two targets in the same pipeline share the same seed. (Even

dynamic branches have different names and thus different seeds.) You can recover the seed of a completed target with tar_meta(your_target, seed) and run `tar_seed_set()` on the result to locally recreate the target's initial RNG state.

path A character string representing a bookdown directory with an 'index.Rmd' file or a character vector of files to be compiled via bookdown.

tidy_eval Logical, whether to enable tidy evaluation when interpreting command and pattern. If TRUE, you can use the "bang-bang" operator !! to programmatically insert the values of global objects.

packages Character vector of packages to load right before the target runs or the output data is reloaded for downstream targets. Use tar_option_set() to set packages globally for all subsequent targets you define.

library Character vector of library paths to try when loading packages.

error Character of length 1, what to do if the target stops and throws an error. Options:

- "stop": the whole pipeline stops and throws an error.
- "continue": the whole pipeline keeps going.
- "abridge": any currently running targets keep running, but no new targets launch after that. (Visit `https://books.ropensci.org/targets/debugging.html` to learn how to debug targets using saved workspaces.)
- "null": The errored target continues and returns NULL. The data hash is deliberately wrong so the target is not up to date for the next run of the pipeline.

deployment Character of length 1. If deployment is "main", then the target will run on the central controlling R process. Otherwise, if deployment is "worker" and you set up the pipeline with distributed/parallel computing, then the target runs on a parallel worker. For more on distributed/parallel computing in targets, please visit `https://books.ropensci.org/targets/crew.html`.

priority Numeric of length 1 between 0 and 1. Controls which targets get deployed first when multiple competing targets are ready simultaneously. Targets with priorities closer to 1 get dispatched earlier (and polled earlier in `tar_make_future()`).

resources Object returned by tar_resources() with optional settings for high-performance computing functionality, alternative data storage formats, and other optional capabilities of targets. See tar_resources() for details.

retrieval Character of length 1, only relevant to `tar_make_clustermq()` and `tar_make_future()`. Must be one of the following values:

- "main": the target's dependencies are loaded on the host machine and sent to the worker before the target runs.
- "worker": the worker loads the targets dependencies.
- "none": the dependencies are not loaded at all. This choice is almost never recommended. It is only for niche situations, e.g. the data needs to be loaded explicitly from another language.

cue An optional object from tar_cue() to customize the rules that decide whether the target is up to date.

description   Character of length 1, a custom free-form human-readable text description of the
                 target. Descriptions appear as target labels in functions like [tar_manifest()](#)
                 and [tar_visnetwork()](#), and they let you select subsets of targets for the names
                 argument of functions like [tar_make()](#). For example, tar_manifest(names =
                 tar_described_as(starts_with("survival model"))) lists all the targets
                 whose descriptions start with the character string "survival model".

...   Named arguments to bookdown::render_book(). These arguments are evalu-
                 ated when the target actually runs in tar_make(), not when the target is defined.

Details

tar_render_book() is an alternative to tar_target() for Bookdown books that depend on other
targets. The folder or files to be rendered via [bookdown::render_book](#) should mention dependency
targets with tar_load() and tar_read() in the active code chunks (which also allows you to
render the report outside the pipeline if the _targets/ data store already exists). (Do not use
tar_load_raw() or tar_read_raw() for this.) Then, tar_render_book() defines a special kind
of target. It 1. Finds all the tar_load()/tar_read() dependencies in any RMarkdown file within
the bookdown directory or supplied files and inserts them into the target's command. This enforces
the proper dependency relationships. (Do not use tar_load_raw() or tar_read_raw() for this.)
2. Sets format = "file" (see tar_target()) so targets watches the files at the returned paths
and reruns the report if those files change. 3. Configures the target's command to return both the
output report files and the input source directory/files. All these file paths are relative paths so the
project stays portable. 4. Forces the report to run in the user's current working directory instead
of the working directory of the report. 5. Sets convenient default options such as deployment =
"main" in the target.

Value

A target object with format = "file". When this target runs, it returns a character vector of file
paths: the rendered document, and the source directory/files. All returned paths are *relative* paths
to ensure portability (so that the project can be moved from one file system to another without
invalidating the target). See the "Target objects" section for background.

Index