

Package: covrpage (via r-universe)

June 21, 2024

Title Create a Readme Summary Page for GitHub testthat and covr
Outputs

Version 0.2

Description Create a README file to place in package test subdirectory
to communicate with other users in more detail results of
testthat and covr.

License MIT + file LICENSE

URL <https://github.com/yonicd/covrpage>

BugReports <https://github.com/yonicd/covrpage/issues>

Encoding UTF-8

LazyData false

Depends R (>= 4.0)

Imports utils, git2r, rmarkdown, covr, stats, testthat, curl, whisker,
withr, remotes

RoxygenNote 7.2.1

Suggests knitr, spelling, rstudioapi

VignetteBuilder knitr

Language en-US

Repository <https://thinkr-open.r-universe.dev>

RemoteUrl <https://github.com/yonicd/covrpage>

RemoteRef HEAD

RemoteSha bafeb195d1999b8c3a48186e26f59cafcdfa3670

Contents

coverage_skip	2
covrpage	3
covrpage_ci	3
covrpage_log	4

covrpage_snapshot	5
covr_log	6
covr_summary	6
make_badge	7
map_test	8
map_testthat	8
map_testthat_snapshot	9
package_name	10
tencrypt	10
testthat_summary	11
use_covrpage	12
use_covrpage_vignette	13
Index	14

coverage_skip	<i>Run covr with failing tests</i>
---------------	------------------------------------

Description

Run `package_coverage` ignoring failing tests from `test_dir`.

Usage

```
coverage_skip(test_path = "tests/testthat", test_skip = NULL)
```

Arguments

<code>test_path</code>	character, path to test directory, Default: 'tests/testthat'
<code>test_skip</code>	data.frame, mapping of failing tests, PARAM_DESCRIPTION, Default: NULL

Details

If `test_skip` is NULL then a new `test_dir` will be run internally checking for failing tests.

See Also

`test_dir` `package_coverage`

Other utility: `covr_summary()`, `map_testthat_snapshot()`, `map_testthat()`, `map_test()`, `package_name()`, `testthat_summary()`

covrpage	<i>Render covrpage</i>
----------	------------------------

Description

Render the template of the covrpage in the package directory.

Usage

```
covrpage(
  path = getwd(),
  preview = TRUE,
  auto_push = FALSE,
  update_badge = TRUE,
  vignette = FALSE
)
```

Arguments

path	path to package
preview	boolean, to open the output in viewer, Default: TRUE
auto_push	boolean, push to remote repo on exit, Default: FALSE
update_badge	boolean, locate badge in README and update with testthat results, Default: TRUE
vignette	boolean, if TRUE then the covrpage README will be copied as a vignette of the package, Default: FALSE

See Also

[render](#)

Other invoke: [covrpage_ci\(\)](#), [covrpage_snapshot\(\)](#)

covrpage_ci	<i>covrpage for travis</i>
-------------	----------------------------

Description

wrapper for covrpage call with switches for running after-success on travis and deploying gh-pages.

Usage

```
covrpage_ci(path = getwd(), ...)
```

Arguments

path character, path to package, Default: getwd()
... arguments to pass to covrpage

See Also

Other invoke: [covrpage_snapshot\(\)](#), [covrpage\(\)](#)

Examples

```
## Not run:  
if(interactive()){  
  covrpage_ci()  
}  
  
## End(Not run)
```

covrpage_log

Query 'git log' for covrpage history

Description

Retrieve log of covrpage README.md files committed to git version control

Usage

```
covrpage_log()
```

Value

list

See Also

Other log: [covr_log\(\)](#)

Examples

```
## Not run:  
covrpage_log()  
  
## End(Not run)
```

covrpage_snapshot *Run covrpage on a GitHub Repository*

Description

Runs a covrpage on a GitHub repository in a sterile environment

Usage

```
covrpage_snapshot(  
  repo,  
  branch = NULL,  
  PAT = Sys.getenv("GITHUB_PAT"),  
  host = "api.github.com",  
  preview = TRUE  
)
```

Arguments

repo	character, Public GitHub repository
branch	character, Name of the branch to checkout. Default: NULL
PAT	character, GitHub Personal Access Token, Default: Sys.getenv("GITHUB_PAT")
host	character, GitHub API host to use, Default: "api.github.com"
preview	boolean, to open the output in viewer, Default: TRUE

Details

Creates a markdown output that is sent to the internal viewer.

Setting branch to NULL means that [clone](#) will use the remote's default branch.

See Also

[covrpage clone install_github](#)

Other invoke: [covrpage_ci\(\)](#), [covrpage\(\)](#)

Examples

```
## Not run:  
if(interactive()){  
  covrpage_snapshot('tidyverse/purrr')  
}  
  
## End(Not run)
```

covr_log	<i>Query 'git log' for code coverage history</i>
----------	--------------------------------------------------

Description

Retrieve log of code coverages evaluated by covrpage that were committed to git version control.

Usage

```
covr_log()
```

Value

data.frame

See Also

Other log: [covrpage_log\(\)](#)

Examples

```
## Not run:  
covr_log()  
  
## End(Not run)
```

covr_summary	<i>Summary outputs for covr object</i>
--------------	----------------------------------------

Description

Creates a summary data.frame for covr object output that mimics the console print method of [package_coverage](#)

Usage

```
covr_summary(x, package, failed = FALSE)
```

Arguments

x	covr object
package	character, package name
failed	flag for failed test, Default: FALSE

Value

data.frame

See Also

[package_coverage](#)

Other utility: [coverage_skip\(\)](#), [map_testthat_snapshot\(\)](#), [map_testthat\(\)](#), [map_test\(\)](#), [package_name\(\)](#), [testthat_summary\(\)](#)

make_badge

Create covrpage Badge

Description

Create covrpage badge for test status and date that is connected to the README.md created in the tests directory.

Usage

```
make_badge(remote_origin = NULL, active_branch = NULL, tinyurl = TRUE)
```

Arguments

`remote_origin` character, username/repo of the remote origin, Default: NULL

`active_branch` character, current active git branch, Default: NULL

`tinyurl` boolean, convert badge uri to tinyurl link, Default: TRUE

Details

if either inputs are NULL then the .git file is inspected to gather the information

Value

character

See Also

Other setup: [tencrypt\(\)](#), [use_covrpage_vignette\(\)](#), [use_covrpage\(\)](#)

map_test	<i>Map the hierarchy structure of a testthat test file</i>
----------	------------------------------------------------------------

Description

Query a test file to see what expectations are defined in it.

Usage

```
map_test(path)
```

Arguments

path character, path to file

Details

Return data.frame containing which expectations are in the test file by context/description/test/expectation/linorange

Value

data.frame

See Also

[getParseData](#)

Other utility: [coverage_skip\(\)](#), [covr_summary\(\)](#), [map_testthat_snapshot\(\)](#), [map_testthat\(\)](#), [package_name\(\)](#), [testthat_summary\(\)](#)

map_testthat	<i>Map the hierarchy structure of testthat directory</i>
--------------	----------------------------------------------------------

Description

Query a testthat directory for the unit test structure.

Usage

```
map_testthat(path = "tests/testthat")
```

Arguments

path character, path to tests, Default: 'tests/testthat'

Details

Return data.frame containing which expectations are in the testthat directory by file/context/description/test/expectation/linorange

Value

data.frame

See Also

[setNames](#)

Other utility: [coverage_skip\(\)](#), [covr_summary\(\)](#), [map_testthat_snapshot\(\)](#), [map_test\(\)](#), [package_name\(\)](#), [testthat_summary\(\)](#)

map_testthat_snapshot *Run map_testthat on a GitHub Repository*

Description

Runs a coverage on a GitHub repository in a sterile environment

Usage

```
map_testthat_snapshot(  
  repo,  
  branch = NULL,  
  PAT = Sys.getenv("GITHUB_PAT"),  
  host = "api.github.com"  
)
```

Arguments

repo	character, Public GitHub repository
branch	character, Name of the branch to checkout. Default: NULL
PAT	character, GitHub Personal Access Token, Default: Sys.getenv("GITHUB_PAT")
host	character, GitHub API host to use, Default: "api.github.com"

Details

Creates a markdown output that is sent to the internal viewer.

Setting branch to NULL means that `clone` will use the remote's default branch.

Value

tibble

See Also

[covrpage](#) [clone](#) [install_github](#)

Other utility: [coverage_skip\(\)](#), [covr_summary\(\)](#), [map_testthat\(\)](#), [map_test\(\)](#), [package_name\(\)](#), [testthat_summary\(\)](#)

Examples

```
## Not run:
if(interactive()){
  map_testthat_snapshot('tidyverse/purrr')
}

## End(Not run)
```

package_name	<i>package name</i>
--------------	---------------------

Description

Retrieve package name

Usage

```
package_name(path = getwd())
```

Arguments

path path to package

See Also

Other utility: [coverage_skip\(\)](#), [covr_summary\(\)](#), [map_testthat_snapshot\(\)](#), [map_testthat\(\)](#), [map_test\(\)](#), [testthat_summary\(\)](#)

tencrypt	<i>Encrypt Travis environment variables</i>
----------	---------------------------------------------

Description

Add encrypted Travis environment variable to git directory and .travis.yml.

Usage

```
tencrypt(
  r_obj = Sys.getenv("GITHUB_PAT"),
  travis_env = "GITHUB_PAT",
  add = TRUE
)
```

Arguments

r_obj	object value to encrypt
travis_env	character, name of Travis environment variable
add	boolean, add directly to .travis.yml, Default: TRUE

See Also

Other setup: [make_badge\(\)](#), [use_covrpage_vignette\(\)](#), [use_covrpage\(\)](#)

testthat_summary	<i>Summary outputs for testthat object</i>
------------------	--------------------------------------------

Description

Summary outputs for testthat object that is used in rmd

Usage

```
testthat_summary(x, type = c("short", "long"))
```

Arguments

x	list, testthat object output
type	character, format of the object output, Default: c("short", "long")

Value

data.frame

See Also

Other utility: [coverage_skip\(\)](#), [covr_summary\(\)](#), [map_testthat_snapshot\(\)](#), [map_testthat\(\)](#), [map_test\(\)](#), [package_name\(\)](#)

 use_covrpage

 Add covrpage to Travis CI for a git directory

Description

Adds bash file to execute after Travis test is successfully completed.

Usage

```
use_covrpage(
  file = ".travis/covrpage.sh",
  travis_type = c("bash", "pkgdown", "tic"),
  travis_blocks = c("after_success", "after_failure", "deploy_ghpages"),
  gh_user = gsub("@(.*)$", "", system("git config user.email", intern = TRUE)),
  push_branch = "test",
  repo = system("git config travis.slug", intern = TRUE),
  deploy_branch = system("git rev-parse --abbrev-ref HEAD", intern = TRUE)
)
```

Arguments

file	character, file path to save bash script to, Default: '.travis/covrpage.sh'
travis_type	character, build .travis.yml template for bash script, pkgdown deployment or using the 'tic' package from ropenscilabs on GitHub Default: c('bash','pkgdown','tic')
travis_blocks	character, vector that defines what .travis.yml blocks tocreate, Default: c('after_success','after_failure','de
gh_user	character, Github user name, Default: gsub("@(.*)\$", "", system("git config user.email", intern = TRUE))
push_branch	character, branch Travis will create to push updated covrpage readme file, Default: 'test'
repo	character, name of git repo that package is hosted on, Default: system("git config travis.slug", intern = TRUE)
deploy_branch	character, name of branch that current package build is located, Default: system("git rev-parse --abbrev-ref HEAD", intern = TRUE) (usually 'master')

See Also

Other setup: [make_badge\(\)](#), [tencrypt\(\)](#), [use_covrpage_vignette\(\)](#)

Examples

```
use_covrpage(travis_type = 'bash',file=file.path(tempdir(),'myfile.sh'))

#does not write to disk
use_covrpage(travis_type = 'bash',file=NULL)

use_covrpage(travis_type = 'pkgdown')
```

```
use_covrpage(travis_type = 'pkgdown', travis_block = 'after_success')
use_covrpage(travis_type = 'pkgdown', travis_block = 'after_failure')
use_covrpage(travis_type = 'pkgdown', travis_block = c('after_success', 'after_failure'))
```

use_covrpage_vignette *Create a vignette for covrpage*

Description

Perform general setup and copying of tests/README.md into vignette subdirectory

Usage

```
use_covrpage_vignette(path = ".")
```

Arguments

path character, path to root of file, Default: '.'

See Also

Other setup: [make_badge\(\)](#), [tencrypt\(\)](#), [use_covrpage\(\)](#)

Examples

```
## Not run:
use_covrpage_vignette()

## End(Not run)
```

Index

- * **invoke**
 - coverpage, [3](#)
 - coverpage_ci, [3](#)
 - coverpage_snapshot, [5](#)
- * **log**
 - cover_log, [6](#)
 - coverpage_log, [4](#)
- * **setup**
 - make_badge, [7](#)
 - tencrypt, [10](#)
 - use_coverpage, [12](#)
 - use_coverpage_vignette, [13](#)
- * **utility**
 - coverage_skip, [2](#)
 - cover_summary, [6](#)
 - map_test, [8](#)
 - map_testthat, [8](#)
 - map_testthat_snapshot, [9](#)
 - package_name, [10](#)
 - testthat_summary, [11](#)

[clone](#), [5](#), [9](#)

[coverage_skip](#), [2](#), [7–11](#)

[cover_log](#), [4](#), [6](#)

[cover_summary](#), [2](#), [6](#), [8–11](#)

[coverpage](#), [3](#), [4](#), [5](#), [9](#)

[coverpage_ci](#), [3](#), [3](#), [5](#)

[coverpage_log](#), [4](#), [6](#)

[coverpage_snapshot](#), [3](#), [4](#), [5](#)

[getParseData](#), [8](#)

[install_github](#), [5](#), [9](#)

[make_badge](#), [7](#), [11–13](#)

[map_test](#), [2](#), [7](#), [8](#), [9–11](#)

[map_testthat](#), [2](#), [7](#), [8](#), [8](#), [9–11](#)

[map_testthat_snapshot](#), [2](#), [7–9](#), [9](#), [10](#), [11](#)

[package_coverage](#), [2](#), [6](#), [7](#)

[package_name](#), [2](#), [7–9](#), [10](#), [11](#)

[render](#), [3](#)

[setNames](#), [9](#)

[tencrypt](#), [7](#), [10](#), [12](#), [13](#)

[test_dir](#), [2](#)

[testthat_summary](#), [2](#), [7–10](#), [11](#)

[use_coverpage](#), [7](#), [11](#), [12](#), [13](#)

[use_coverpage_vignette](#), [7](#), [11](#), [12](#), [13](#)