Package: kwb.pkgstatus (via r-universe)

September 15, 2024

Title R package for checking KWB package status

Version 0.3.5

Description R package for checking KWB package status (e.g. generating https://kwb-r.github.io/status).

License MIT + file LICENSE

URL https://github.com/KWB-R/kwb.pkgstatus

BugReports https://github.com/KWB-R/kwb.pkgstatus/issues

Imports data.table, dplyr, fs, gh, httr, jsonlite, kwb.pkgbuild, lubridate, magrittr, stringr

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

VignetteBuilder knitr

Remotes github::kwb-r/kwb.pkgbuild

ByteCompile true

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

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

RemoteUrl https://github.com/KWB-R/kwb.pkgstatus

RemoteRef HEAD

RemoteSha 84c6b32b8fc68e91b634ddc1f7dcbdcd4ab9d84e

Contents

`badge_appveyor` *badge_appveyor*

Description

badge_appveyor

Usage

```
badge_appveyor(repo_full_names)
```

Arguments

`repo_full_names`
: vector with combination of username/repo (e.g. c("KWB-R/kwb.utils", "KWB-R/kwb.db"))

Value

appveyor badges for provided repo_full_names

badge_codecov *badge_codecov*

Description

badge_codecov

Usage

```
badge_codecov(repo_full_names)
```

Arguments

repo_full_names	vector with combination of username/repo (e.g. c("KWB-R/kwb.utils", "KWB-R/kwb.db"))

Value

codecov badges for provided repo_full_names

badge_cran *badge_cran*

Description

badge_cran

Usage

```
badge_cran(repo_names)
```

Arguments

repo_names	vector of repository names (e.g. c("kwb.utils", "kwb.db"))

Value

crank badges for provided repo_names

badge_dependencies *badge_dependencies*

Description

badge_dependencies

Usage

```
badge_dependencies(repo_names)
```

Arguments

repo_names	vector of repository names (e.g. c("kwb.utils", "kwb.db"))

Value

dependency badges for provided repo_names

badge_gitlab *badge_gitlab*

Description

badge_gitlab

Usage

```
badge_gitlab(
  url,
  logo_path = paste0("https://gitlab.com/gitlab-com/gitlab-artwork/raw/",
    "master/logo/logo-square.png"),
  size = 24
)
```

Arguments

url	url to repository on Gitlab
logo_path	path to Gitlab logo (default: "https://gitlab.com/gitlab-com/gitlab-artwork/raw/master/logo/logo-square.png")
size	size of logo in pixels (default: 24)

Value

Gitlab logo in html with path to repository in Gitlab

badge_license *badge_license*

Description

badge_license

Usage

```
badge_license(license_keys, github_token = Sys.getenv("GITHUB_TOKEN"))
```

Arguments

license_keys	one or many valid license keys from c("agpl-3.0", "apache-2.0", "bsd-2-clause", "bsd-3-clause", "epl-2.0", "gpl-2.0", "gpl-3.0", "lgpl-2.1", "lgpl-3.0", "mit", "mpl-2.0", "unlicense")
github_token	github access token (default: Sys.getenv("GITHUB_TOKEN"))

Value

badge for all provided license keys

---

badge_opencpu *badge_opencpu*

Description

badge_opencpu

Usage

```
badge_opencpu(
  url,
  logo_path = "https://avatars2.githubusercontent.com/u/28672890?s=200&v=4",
  size = 24
)
```

Arguments

url	url to repository on Gitlab
logo_path	path to OpenCpu logo (default: "https://avatars2.githubusercontent.com/u/28672890?s=200&v=4")
size	size of logo in pixels (default: 24)

Value

OpenCpu logo in html with path to R package on OpenCpu

badge_travis *badge_travis*

Description

badge_travis

Usage

```
badge_travis(repo_full_names)
```

Arguments

repo_full_names	vector with combination of username/repo (e.g. c("KWB-R/kwb.utils", "KWB-R/kwb.db"))

Value

travis badges for provided repo_full_names

badge_zenodo *badge_zenodo*

Description

badge_zenodo

Usage

```
badge_zenodo(repo_full_names, zenodo_token = Sys.getenv("ZENODO_TOKEN"))
```

Arguments

repo_full_names	vector with combination of username/repo (e.g. c("KWB-R/kwb.utils", "KWB-R/kwb.db"))
zenodo_token	zenodo authentication token (default: Sys.getenv("ZENODO_TOKEN"))

Value

zenodo badges for provided repo_full_names

check_docu_dev *Check documentation: development*

Description

Check documentation: development

Usage

```
check_docu_dev(repo_names, url = "http://kwb-r.github.io")
```

Arguments

repo_names	vector of repository names to be checked
url	main url for Github pages (default: "http://kwb-r.github.io")

Value

character vector with links in case documentation of development version for R packages is available

check_docu_release *Check documentation: release*

Description

Check documentation: release

Usage

```
check_docu_release(repo_names, url = "http://kwb-r.github.io")
```

Arguments

repo_names	vector of repository names to be checked
url	main url for Github pages (default: "http://kwb-r.github.io")

Value

character vector with links in case documentation of latest release for R packages is available

`check_gitlab_backup` *check_gitlab_backup*

Description

check_gitlab_backup

Usage

```
check_gitlab_backup(
  group = "KWB-R",
  github_token = Sys.getenv("GITHUB_TOKEN"),
  gitlab_token = Sys.getenv("GITLAB_TOKEN")
)
```

Arguments

`group`	username or organisation for Github/Gitlab (default: "KWB-R")
`github_token`	github access token (default: Sys.getenv("GITHUB_TOKEN"))
`gitlab_token`	gitlab access token (default: Sys.getenv("GITLAB_TOKEN")))

Value

data.frame containing all Github repositoriers that are mirrored in Gitlab (i.e. were at least syncronised within the last 2 hours)

`check_opencpu_deploy` *check_opencpu_deploy: get all Github repos that are deployed on OpenCpu*

Description

Direct deployment of R packages (including vignette build) by using webhooks as described in OpenCpu blog post (https://www.opencpu.org/posts/opencpu-release-1-4-5/) and online help (https://www.opencpu.org/cloud

Usage

```
check_opencpu_deploy(group = "KWB-R")
```

Arguments

`group`	username or organisation for Github (default: "KWB-R")

Value

data.frame containing names with OpenCpu badges for all Github repositories that are deployed on OpenCpu (default: https://kwb-r.ocpu.io)

`create_report_rpackages` *Create R packages status report#'*

Description

Create R packages status report#'

Usage

```
create_report_rpackages(
  secrets_csv = NULL,
  non_r_packages = get_non_r_packages(),
  export_dir = ".",
  input_rmd = system.file("extdata/reports/status_report.Rmd", package =
    "kwb.pkgstatus")
)
```

Arguments

`secrets_csv`	path to "secrets.csv" file, if "NULL" Sys.env variables for the following services are used/need to be defined: APPVEYOR_TOKEN, GITHUB_TOKEN, GITLAB_TOKEN, CODECOV_TOKEN, ZENODO_TOKEN, (default: NULL)
`non_r_packages`	a character vector with repositories in KWB-R group that are not R packages (default: `get_non_r_packages`)
`export_dir`	report export directory (default: ".")
`input_rmd`	default: system.file("extdata/reports/status_report.Rmd", package = "kwb.pkgstatus")

Value

creates html status report for R packages and returns the absolute path to the export directory

`get_coverage` *get_coverage*

Description

get_coverage

Usage

```
get_coverage(
  repo_full_name,
  codecov_token = Sys.getenv("CODECOV_TOKEN"),
  dbg = TRUE
)
```

Arguments

repo_full_name	one combination of username/repo (e.g."KWB-R/kwb.db")
codecov_token	codecov authentication token (default: Sys.getenv("CODECOV_TOKEN"))
dbg	debug if TRUE (default: TRUE)

Value

codecov coverage in percent for provided repo_full_name

---

get_coverages *get_coverage*

Description

get_coverage

Usage

```
get_coverages(
  repo_full_names,
  codecov_token = Sys.getenv("CODECOV_TOKEN"),
  dbg = TRUE
)
```

Arguments

repo_full_names	vector with combination of username/repo (e.g. c("KWB-R/kwb.utils", "KWB-R/kwb.db"))
codecov_token	zenodo authentication token (default: Sys.getenv("CODECOV_TOKEN")
dbg	debug if TRUE (default: TRUE)

Value

data.frame with coverage percent and url for all provided repo_full_names

get_github_repos *get_github_repos*

Description

get_github_repos

Usage

```
get_github_repos(group = "KWB-R", github_token = Sys.getenv("GITHUB_TOKEN"))
```

Arguments

group	username or organisation for Github (default: "KWB-R")
github_token	github access token (default: Sys.getenv("GITHUB_TOKEN"))

Value

data.frame with for all repositories of the user/organisation defined in parameter group (private repos will only be accessible if the token is configured to allow that)

get_gitlab_repos *get_gitlab_repos*

Description

get_gitlab_repos

Usage

```
get_gitlab_repos(group = "KWB-R", gitlab_token = Sys.getenv("GITLAB_TOKEN"))
```

Arguments

group	username or organisation for Gitlab (default: "KWB-R")
gitlab_token	gitlab access token (default: Sys.getenv("GITLAB_TOKEN"))

Value

data.frame with for all repositories of the user/organisation defined in parameter group (private repos will only be accessible if the token is configured to allow that)

get_non_r_packages *Helper function: get_non_r_packages*

Description

Helper function: get_non_r_packages

Usage

```
get_non_r_packages()
```

Value

returns vector with KWB-R repos on Github, which are not R packages

Examples

```
get_non_r_packages()
```

prepare_status_rpackages *prepare_status_rpackages*

Description

prepare_status_rpackages

Usage

```
prepare_status_rpackages(
  secrets_csv = NULL,
  non_r_packages = get_non_r_packages()
)
```

Arguments

secrets_csv	path to "secrets.csv" file, if "NULL" Sys.env variables for the following services are used/need to be defined: APPVEYOR_TOKEN, GITHUB_TOKEN, GITLAB_TOKEN, CODECOV_TOKEN, ZENODO_TOKEN, (default: NULL)
non_r_packages	a character vector with repositories in KWB-R group that are not R packages (default: get_non_r_packages)

Value

data.frame with R package status information

process_hitter_response *Helper function for Zenodo*

Description

Helper function for Zenodo

Usage

```
process_hitter_response(response)
```

Arguments

response	response provided by httr::content()

Value

a tibble of available Zenodo data

url_success *url_success*

Description

url_success

Usage

```
url_success(url)
```

Arguments

url	url of documentation website

Value

TRUE in case HTTP status code is 200, if not: FALSE

`zen_collections` *Zenodo: get available collections*

Description

Zenodo: get available collections

Usage

```
zen_collections(n = 1000, access_token = Sys.getenv("ZENODO_TOKEN"))
```

Arguments

`n`	number of zenodo entries ("size") to return per API call (default: 1000)
`access_token`	Zenodo access token (default: Sys.getenv("ZENODO_TOKEN"))

Value

a tibble of available Zenodo data

See Also

https://developers.zenodo.org/#depositions

Index