

Package: pkgpub (via r-universe)

May 31, 2026

Type Package
Title Publish Packages
Version 0.4.0
Description Publish packages to cran-like repos.
Encoding UTF-8
RoxygenNote 7.3.2
Imports magrittr, desc, drat, pkgbuild, withr, gert, glue, rlang
License MIT + file LICENSE
Suggests testthat (>= 3.0.0)
URL <https://a2-ai.github.io/pkgpub/>
Config/testthat/edition 3
Config/pak/sysreqs git libgit2-dev libssl-dev
Repository <https://a2-ai.r-universe.dev>
Date/Publication 2025-04-03 13:17:30 UTC
RemoteUrl <https://github.com/a2-ai/pkgpub>
RemoteRef HEAD
RemoteSha fa3e20a411847998ecc4fc6f5be7ee5608b4a664

Contents

build_pkg	2
create_tagged_repo	3
git_url	3
head_tag	3
insert_packages	4
new_cranlike_repo	4
skip_remaining_if_dev_tag	5
tag_version	5

Index	6
--------------	----------

build_pkg	<i>build various versions of a package</i>
-----------	--

Description

build various versions of a package

Usage

```
build_pkg(
  .pkgdir = ".",
  types = c("source", "binary"),
  repository = NULL,
  origin = NULL,
  addl_meta = NULL,
  supplement_version = FALSE,
  overwrite = TRUE,
  remove_remotes = FALSE,
  ...
)
```

Arguments

.pkgdir	path to package
types	types of package to build
repository	repository name being built for
origin	package source
addl_meta	additional metadata
supplement_version	add additional version info (unix timestamp) to version. TRUE inspects the pkg folder as a git repo, also may provide a character or numeric value to append
overwrite	overwrite fields already present when adding fields
remove_remotes	delete the remotes field
...	parameters to pass to pkgbuild

Details

supplementing version can be done whenever a build occurs that does not correspond to a formal release/tag. This will automatically add information about the git hash (if available), as well as increment the version number with a unix timestamp that corresponds to the last git hash (if present) or the current system time, if git is not present and no version timestamp is provided.

removing the remotes field can be helpful when dealing with consolidated package installs where all packages are available in a single repository. For example, pulling everything into a posit package manager mirror.

create_tagged_repo *create a cranlike repo that contains the tagged package*

Description

create a cranlike repo that contains the tagged package

Usage

```
create_tagged_repo(.dir, repo_name = "pkgpub", ...)
```

Arguments

.dir	directory to create the repo in
repo_name	repository name to specify in the description
...	additional arguments passed to build_pkg

git_url *get the origin url*

Description

get the origin url

Usage

```
git_url(remote = "origin")
```

Arguments

remote	remote name
--------	-------------

head_tag *get the tag of the head (current commit)*

Description

get the tag of the head (current commit)

Usage

```
head_tag()
```

insert_packages	<i>insert packages into a drat repo</i>
-----------------	---

Description

insert packages into a drat repo

Usage

```
insert_packages(pkgs, repository, archive = TRUE, ..., os_flavor = NULL)
```

Arguments

pkgs	list of packages to insert
repository	repository folder to insert
archive	whether to archive the packages after insertion to prevent duplicates
...	parameters passed to 'drat::archivePackages'
os_flavor	optional os_flavor specification, defaults to NULL unless detected as aarch64

Details

repository folder should most likely correspond to the name of the repository set in the pkg meta-data, however realistically, the time that they need to match when repos is set in a user session in the 'getOption('repos')' field

new_cranlike_repo	<i>new cranlike repo will create a cranlike repo</i>
-------------------	--

Description

new cranlike repo will create a cranlike repo

Usage

```
new_cranlike_repo(dir = getwd())
```

Arguments

dir	directory to create
-----	---------------------

skip_remaining_if_dev_tag
skip remaining drone CI steps if tag is dev tag

Description

skip remaining drone CI steps if tag is dev tag

Usage

skip_remaining_if_dev_tag()

tag_version *add a git tag based on the pkg version*

Description

add a git tag based on the pkg version

Usage

tag_version(.dir = getwd(), .dirty = FALSE, .fetch = TRUE, .push = TRUE)

Arguments

.dir	dir of repo
.dirty	whether to tag given the repo is in a dirty state
.fetch	run a git fetch first to get tags
.push	push created tag

Index

`build_pkg`, 2

`create_tagged_repo`, 3

`git_url`, 3

`head_tag`, 3

`insert_packages`, 4

`new_cranlike_repo`, 4

`skip_remaining_if_dev_tag`, 5

`tag_version`, 5