Package ‘ghtravis’

June 29, 2017

Type Package

Title Interfaces with GitHub API and Travis

Version 0.5.5

Author John Muschelli

Maintainer John Muschelli <muschellij2@gmail.com>

Description Interfaces and compares things on 'GitHub'
'API' and 'Travis' 'CI'.

Imports utils, devtools, yaml, httr

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests covr

NeedsCompilation no

R topics documented:

`all_remote_package_deps` *Get Remote Package Dependencies*

Description

Get Remote Package Dependencies

Usage

```
all_remote_package_deps(path = "DESCRIPTION", remotes = NULL,
  verbose = TRUE, ..., dependencies = c("Depends", "Imports", "LinkingTo",
  "Suggests"), exclude_remotes = TRUE)

remote_package_deps(..., dependencies = c("Depends", "Imports", "LinkingTo",
  "Suggests"))

missing_remote_deps(path = "DESCRIPTION", verbose = TRUE, ...)

install_missing_remote_deps(path = "DESCRIPTION", ...)
```

Arguments

`path`	Path to DESCRIPTION file
`remotes`	Repository if one specific
`verbose`	Print diagnostic message
`...`	arguments passed to `get_remote_package_dcf`
`dependencies`	List of dependencies to parse, passed to `get_dep_table`
`exclude_remotes`	exclude Remotes in dependency table

Value

A `data.frame` of Dependencies (`all_remote_package_deps`) or a list of a package dependency table and a vector of package names for the remotes (`remote_package_deps`)

Examples

```
remote_package_deps("stnava/ANTsRCore")
path = example_description_file()
all_remote_package_deps(path = path)
all_remote_package_deps(remotes = c("stnava/ANTsRCore", "stnava/ITKR"))
missing_remote_deps(path = path)
```

binary_release_table *Get Binary Release Table for a repository*

Description

Parses a remote and gets a table of binary releases

Usage

```
binary_release_table(repo, pat = NULL, verbose = TRUE, ...)

binary_table_no_tags(repo, pat = NULL, ...)
```

Arguments

repo	Remote repository name
pat	GitHub Personal Authentication Token (PAT)
verbose	print diagnostic messages
...	additional arguments to GET

Value

data.frame of binary releases

Examples

```
repo = "stnava/ANTsR"
binary_release_table(repo)
binary_release_table("muschellij2/ANTsR")
repo = "stnava/ANTsR"
binary_table_no_tags(repo)
binary_table_no_tags("muschellij2/ANTsR")
```

dcf_package_version *Get the version of the package from the DESCRIPTION file*

Description

Extracts Package version from DESCRIPTION file

Usage

```
dcf_package_version(path = "DESCRIPTION")
```

Arguments

path	Path to DESCRIPTION file

Value

Return the Package Version

Examples

```
path = example_description_file()
desc = dcf_package_version(path)
```

deployed_tarball_version *Deployed Tarball Version*

Description

Downloads a source tarball from GitHub and parses the version in that source

Usage

```
deployed_tarball_version(repo = "stnava/ITKR", tag = "latest")

deployed_tarball(repo = "stnava/ITKR", tag = "latest", version = NULL)
```

Arguments

repo	GitHub repository name
tag	Tag for the tarball
version	version of package tarball if applicable

Value

Version number character vector

Examples

```
deployed_tarball_version(  repo = "stnava/ITKR", tag = "latest")
```

deploy_travis *Rewrite Travis for Deployment*

Description

Writes the Deployment part of the Travis YAML for a binary package

Usage

```
deploy_travis(api_key, path = "DESCRIPTION", travis_file = ".travis.yml",
  user = "neuroconductor")
```

Arguments

api_key	API Key to Travis Key, Secure
path	Path to DESCRIPTION file
travis_file	Travis File
user	GitHub username of repository

Value

Name of travis file

Examples

```
travis_file = example_travis_file()
path = example_description_file()
outfile = deploy_travis(api_key = "test",
path = path,
travis_file = travis_file)
readLines(outfile)
```

drop_remotes *Drop Remotes from DESCRIPTION*

Description

Reads a DESCRIPTION file, drops the remotes, and then rewrites the DESCRIPTION file

Usage

```
drop_remotes(path = "DESCRIPTION", drop_remotes = NULL, reorder = FALSE,
  verbose = TRUE, ...)
```

Arguments

path	Path to DESCRIPTION file
drop_remotes	Packages for Remotes to Drop
reorder	Should remotes be reordered after running
verbose	Print diagnostic message
...	if reorder = TRUE then arguments passed to reorder_remotes

Value

Rewrites the DESCRIPTION file without remotes

Examples

```
path = example_description_file()
x = drop_remotes(path, reorder = TRUE)
readLines(x)
```

example_description_file

Example DESCRIPTION File

Description

Copies example DESCRIPTION file for parsing

Usage

```
example_description_file()

example_travis_file()
```

Value

Path of DESCRIPTION file

Examples

```
example_description_file()
```

get_dep_table *Get Dependency Table*

Description

Get a Table of the Dependencies

Usage

```
get_dep_table(path = "DESCRIPTION", dependencies = c("Depends", "Imports",
  "LinkingTo", "Suggests"), limit_compare = FALSE)
```

Arguments

`path`	Path to DESCRIPTION file
`dependencies`	List of dependencies to parse
`limit_compare`	only keep rows where a comparison operator is found

Value

A `data.frame` of

name name of the package

compare comparison operator

version version of the package

Examples

```
path = example_description_file()
get_dep_table(path = path)
```

get_remotes *Get the Remotes from a Package*

Description

Parse a DESCRIPTION file and returns the remotes

Usage

```
get_remotes(path = "DESCRIPTION")
```

Arguments

`path`	Path to DESCRIPTION file

Value

Character vector of remotes

Examples

```
path = example_description_file()
get_remotes(path)
```

get_remote_info *Get Remote Information*

Description

Parses DESCRIPTION file

Usage

```
get_remote_info(path = "DESCRIPTION")

remote_package_names(path = "DESCRIPTION")
```

Arguments

path	Path to DESCRIPTION file

Value

List of parsed files

Examples

```
path = example_description_file()
desc = get_remote_info(path)
```

get_remote_package_dcf *Read Remote Package DESCRIPTION file*

Description

Read Remote Package DESCRIPTION file

Usage

```
get_remote_package_dcf(remotes, pat = NULL, url = "https://github.com", ...)

remote_package_dcf(...)
```

Arguments

remotes	Repository name
pat	GitHub Personal Authentication Token (PAT)
url	URL for repo - github
...	not used

Value

List of Package/Version DCF information

Examples

```
get_remote_package_dcf("stnava/ANTsRCore")
remote_package_dcf("stnava/ANTsRCore")
```

`install_remotes_no_dep` *Returns the specific package from Remotes*

Description

Subsets the remotes based on a package

Usage

```
install_remotes_no_dep(path = "DESCRIPTION", package = NULL,
  reorder = TRUE, drop = TRUE, verbose = TRUE, ...)
```

Arguments

`path`	Path to DESCRIPTION file
`package`	Package to subset. If NULL, then all will be installed
`reorder`	should remotes be reordered before running
`drop`	Remotes should be dropped after installing
`verbose`	Print diagnostic message
`...`	arguments to pass to install_github

Value

Character vector of remotes

Output of `install_github`, indicator of TRUE/FALSE for installation

Examples

```
## Not run:
  install_remotes_no_dep()

## End(Not run)
```

install_remote_binaries

Install Binaries from Remotes

Description

Gets Remotes, gets Binaries form GH, and installs them, then drops them from Remotes field

Usage

```
install_remote_binaries(path = "DESCRIPTION", remotes = NULL,
  package = NULL, verbose = TRUE)
```

Arguments

path	Path to DESCRIPTION file passed to remote_binaries
remotes	Remotes to get binaries for - in case not going from DESCRIPTION, passed to remote_binaries
package	Specific package to install
verbose	Print diagnostic messages

Value

Invisible NULL

Examples

```
## Not run:
path = example_description_file()
install_remote_binaries(path = path)

## End(Not run)
```

latest_release_with_binary

Get the Latest Binary for a repository

Description

Parses a remote and gets a URL for a binary release

Usage

```
latest_release_with_binary(repo, pat = NULL, verbose = TRUE, ...)
```

Arguments

repo	Remote repository name
pat	GitHub Personal Authentication Token (PAT)
verbose	print diagnostic messages
...	additional arguments to GET

Value

URL of the binary

Examples

```
repo = "stnava/ANTsR"
latest_release_with_binary(repo)
```

---

make_full_version *Make a vector of version numbers into full versions for comparison*

Description

The `compareVersion` function requires full version numbers for sufficient comparion

Usage

```
make_full_version(x)
```

Arguments

`x`	character vector of version numbers

Value

Character vector of full version numbers

Examples

```
x = c("0.15", "0.15.0")
compareVersion(x[1], x[2])
full = make_full_version(x)
compareVersion(full[1], full[2])
```

---

package_comparison_table *Get Comparison Table from Dependencies*

Description

Gets Remotes and compares to installed packages

Usage

```
package_comparison_table(path = "DESCRIPTION", dependencies = c("Depends",
  "Imports", "LinkingTo"))

update_these_packages(...)
```

Arguments

`path`	Path to DESCRIPTION file passed to `remote_binaries`
`dependencies`	List of dependencies to parse
`...`	not used

Value

Invisible NULL

Examples

```
path = example_description_file()
update_these_packages(path = path)
```

`package_name` *Get Package name*

Description

Get Package name

Usage

```
package_name(path = "DESCRIPTION")
```

Arguments

`path`	Path to DESCRIPTION file

Value

Name of Packge

Examples

```
path = example_description_file()
package_name(path)
```

`package_needs_update` *Check Package Versions*

Description

Looks in installed packages to see if a new version is needed

Usage

```
package_needs_update(version, package)

set_update_var(...)
```

Arguments

`version`	Version of package
`package`	Package to check
`...`	not used

Value

Output from `compareVersion`

`parse_one_remote` *Parse a Remote file*

Description

Parse Remote from DESCRIPTION

Usage

```
parse_one_remote(x)

parse_remotes(x)
```

Arguments

`x`	Parses a remote field

Value

A repo object, from `devtools`

Examples

```
parse_one_remote("stnava/ANTsR")
```

read_dcf *Read DESCRIPTION file*

Description

Reads a DESCRIPTION file

Usage

```
read_dcf(path = "DESCRIPTION")
```

Arguments

path	path to the DESCRIPTION file

Value

List of the fields and the data.frame of the description information

remote_binaries *Get Binaries from Remotes*

Description

Gets Remotes, gets Binaries form GH

Usage

```
remote_binaries(remotes = NULL, path = "DESCRIPTION", package = NULL,
  verbose = TRUE)
```

Arguments

remotes	Remotes to get the binaries for
path	Path to DESCRIPTION field
package	packages to subset if necessary
verbose	print diagnostic messages

Value

List of remote binaries

Examples

```
remote_binaries(remotes = "muschellij2/neurobase")
path = example_description_file()
if (file.exists(path)) {
   remote_binaries(path = path)
}
```

`remote_order` *Order of the Remotes for Installation*

Description

Order of the Remotes for Installation

Usage

```
remote_order(path = "DESCRIPTION", ..., max_iter = 200)

reorder_remotes(path = "DESCRIPTION", verbose = TRUE, ...)
```

Arguments

`path`	Path to DESCRIPTION file
`...`	arguments passed to `remote_package_deps`
`max_iter`	Number of iterations - shouldn't have to change, stops from infinite loop
`verbose`	Print diagnostic message

Value

Order of remotes

Examples

```
path = example_description_file()
x = remote_order(path = path)
x
```

`rewrite_remotes` *Rewrite DESCRIPTION file overwriting Remotes*

Description

Rewrite DESCRIPTION file overwriting Remotes

Usage

```
rewrite_remotes(path, remotes = NULL)
```

Arguments

`path`	Path to DESCRIPTION file
`remotes`	Packages for remotes field

Value

Name of path

`split_remotes` *Remotes Splitter*

Description

Wrapper for splitting and trimming on spaces

Usage

```
split_remotes(x)
```

Arguments

`x`	Character vector of inputs

Value

Character vector with multiple elements

Examples

```
split_remotes(c("stnava/ANTsR, blah/package@asdf"))
```

`subset_remote` *Returns the specific package from Remotes*

Description

Subsets the remotes based on a package

Usage

```
subset_remote(path = "DESCRIPTION", package = "ANTsR")
```

Arguments

`path`	Path to DESCRIPTION file
`package`	Package to subset

Value

Character vector of remotes

Examples

```
path = example_description_file()
desc = get_remote_info(path)
```

sys_ext *System Extension*

Description

Gets System Extension for tarball

Usage

```
sys_ext()
```

Value

Extension for the operating system

Examples

```
sys_ext()
```

tag_table *Tag Table*

Description

Parses a remote and gets a table of tags

Usage

```
tag_table(repo, pat = NULL, ...)
```

Arguments

repo	Remote repository name
pat	GitHub Personal Authentication Token (PAT)
...	additional arguments to GET

Value

data.frame of tags

Examples

```
repo = "stnava/ANTsR"
tag_table(repo)
tag_table("muschellij2/ANTsR")
tag_table("cran/psych@084bdd0ae2630cf31c26d97a6e13e59d3f0f66e6")
```

Index