

Package: inteRgrate (via r-universe)

September 12, 2024

Type Package

Title Opinionated Package Coding Styles

Version 1.0.22

Description A set of functions to enforce styling in functions. This package would typically be used in with a CI platform, such as GitHub and GitLab.

License GPL-2 | GPL-3

Imports cli, glue, lintr (>= 2.0.0), rcmdcheck, remotes, stringr, tibble, usethis

Suggests roxygen2, testthat, withr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Repository <https://jumpingrivers.r-universe.dev>

RemoteUrl <https://github.com/jumpingrivers/inteRgrate>

RemoteRef HEAD

RemoteSha 174bfe4c325a8f6c6ddc36b5e776f6cb9011f42b

Contents

build_pkg	2
check_all	2
check_file_permissions	3
check_gitignore	4
check_lintr	4
check_namespace	4
check_news	5
check_pkg	5
check_readme	6
check_r_filenames	6
check_tidy_description	7

check_version	7
check_via_env	8
create_tag	9
get_default_rproj	9
gitlab_instead_of	10
git_pre_commit	10
set_renvirion_var	11

Index 12

build_pkg	<i>Build R tar gz file</i>
-----------	----------------------------

Description

Builds and sets the PKG_TARBALL & PKG_TARBALL_PATH variables.

Usage

```
build_pkg()
```

```
get_pkg_tar_ball()
```

check_all	<i>Run all checks</i>
-----------	-----------------------

Description

This function will run all checks within the package. See the corresponding check functions for details.

Usage

```
check_all(
  pkg = TRUE,
  lintr = TRUE,
  namespace = TRUE,
  r_filenames = TRUE,
  version = TRUE,
  gitignore = TRUE,
  tidy_description = TRUE,
  readme = TRUE,
  file_permissions = TRUE,
  line_breaks = TRUE,
  tag = TRUE,
  news = TRUE,
  rproj = TRUE
)
```

Arguments

pkg	Build and check package. See details for further information.
lintr, namespace, r_filenames, version, gitignore, readme, tidy_description	Checks
file_permissions, line_breaks	Windows related checks.
tag	Create a tagged release.
news	Check the top line of NEWS.md
rproj	Runs check_rproj

Details

The arguments for the function correspond to a particular check, e.g. `check_ARGNAME()`. Note: As this package matures, this function will include the newer checks.

check_file_permissions

Detects standard Windows related issues

Description

This check tests for windows line breaks and file permissions. It ensures that the file is not executable (txt|md|Rmd|yml|jsonl).

Usage

```
check_file_permissions(repo_files = NULL, path = ".")
```

Arguments

repo_files	By default, we use git to determine the files in the repo. By a vector of files can be passed instead.
path	Location of the package

Examples

```
check_file_permissions()
```

check_gitignore *Check gitignore file*

Description

Checks the gitignore files for recommended patterns. If any of these patterns are missing, an error is raised.

Usage

```
check_gitignore(path = ".")
```

Arguments

path Location of the package

check_lintr *The lintr check*

Description

Runs lint from the lintr package. Also scans for all .Rmd and .R files in other directories.

Usage

```
check_lintr(path = ".")
```

Arguments

path Location of the package

check_namespace *Namespace checks*

Description

Importing too many packages can cause bugs due to namespace clashes. This check enforces a set number of imports. As a rule of thumb, if you only use a few functions from a package, use `importFrom` instead of `import` in your `NAMESPACE` file.

Usage

```
check_namespace(no_imports = NULL, path = ".")
```

Arguments

no_imports	Default NULL. Number of package imports via <code>import()</code> allowed in the NAMESPACE. If NULL, checks for environment variable <code>NO_IMPORTS</code> , otherwise 0.
path	Location of the package

 check_news

Check NEWS format

Description

Checks that NEWS.md exists and title lines follow the correct format. If the pattern is NULL, then the expected format is either: `# pkg_name Version _YYYY-MM-DD_` or `# pkg_name (development version)`

Usage

```
check_news(pattern = NULL, path = ".")
```

```
check_all_news(pattern = NULL)
```

Arguments

pattern	Regular expression NEWS title.
path	Location of the package

Details

The `check_all_news()` function isn't used in the CI. It's a convenient command line function to quickly assess your NEWS.md file

 check_pkg

Check R package

Description

Path is the CI project directory or `..`. Installs and checks the R package. The number of notes and warnings is specified by environmental variables. Default number of Notes/Warnings is 0.

Usage

```
check_pkg(path = ".")
```

Arguments

path	Location of the package
------	-------------------------

check_readme

Check the README.Rmd

Description

If README.Rmd exists, the timestamp will be compared to README.md. An error will be raised if README.md is a newer file.

Usage

```
check_readme(path = ".")
```

Arguments

path Location of the package

Examples

```
check_readme()
```

check_r_filenames

Check R/ filenames

Description

Checks that all filenames in the directory R/ are lower case and the file extension matches .extension

Usage

```
check_r_filenames(path = ".", extension = "R")
```

Arguments

path Location of the package
extension Preferred R file extension. Default R.

Examples

```
check_r_filenames()
```

check_tidy_description
Check tidy description

Description

Checks for a tidy description file (via the usethis function `use_tidy_description`). It also checks that the version numbers conform to the tidy format - X.Y.Z or X.Y.Z.9ABC

Usage

```
check_tidy_description(path = ".")
```

Arguments

path	Location of the package
------	-------------------------

check_version *Version check*

Description

Check if the package version has been updated compared to the a master repo (default is origin/master). This ensures that when branches are merged the package version is bumped.

Usage

```
check_version(repo = NULL, path = ".")
```

Arguments

repo	Default origin/master or origin/main. The repo to compare against.
path	Location of the package

Details

Files listed in `.Rbuildignore` don't count as changes.

Technically we only check for a change in the Version line of the DESCRIPTION file, not an actual version increase.

check_via_env	<i>Run all checks</i>
---------------	-----------------------

Description

This function will run all checks within the package. See the corresponding check functions for details. When `default = "true"`, all checks will run unless the associated environment variable is `"false"`.

Usage

```
check_via_env(
  pkg = NULL,
  lintr = NULL,
  namespace = NULL,
  r_filenames = NULL,
  version = NULL,
  gitignore = NULL,
  tidy_description = NULL,
  readme = NULL,
  file_permissions = NULL,
  line_breaks = NULL,
  tag = NULL,
  news = NULL,
  rproj = NULL,
  default = FALSE
)
```

Arguments

<code>pkg</code>	Build and check package. See details for further information.
<code>lintr</code> , <code>namespace</code> , <code>r_filenames</code> , <code>version</code> , <code>gitignore</code> , <code>readme</code> , <code>tidy_description</code>	Default NULL.
<code>file_permissions</code> , <code>line_breaks</code>	Windows related checks.
<code>tag</code>	Default NULL. Create a tagged release.
<code>news</code>	Check the top line of NEWS.md
<code>rproj</code>	Check Rproject file
<code>default</code>	Default FALSE. The default value the environment variable should take if missing.

Details

The arguments for the function correspond to a particular check, e.g. `check_ARGNAME()`. By default, all arguments are NULL and hence run. However, a value of FALSE or changing the corresponding environment variable `INTERGRATE_ARGNAME` can turn off the check, e.g. `INTERGRATE_NAMESPACE`.

On GitLab you can set environment variables at an organisation level.

Note: As this package matures, this function will include the newer checks.

create_tag	<i>Auto-tagging via CI</i>
------------	----------------------------

Description

Automatically tag the commit via the version number. This requires the environment variable `GITHUB_TOKEN` that has write permission.

Usage

```
create_tag(branch = get_origin_name(), in_development = FALSE)
```

Arguments

`branch` The branch where the tagging will occur. Default master or main.
`in_development` Logical default FALSE.

Details

If the version contains an in development component (e.g. `X.Y.Z.9001`), by default a tag isn't created.

get_default_rproj	<i>Check the .Rproj file</i>
-------------------	------------------------------

Description

If a `.Rproj` file exists, then the values should correspond to those given by `get_default_rproj`. If more than one `.Rproj` file exists, an error is raised.

Usage

```
get_default_rproj()

check_rproj(default_rproj = get_default_rproj(), path = ".")
```

Arguments

default_rproj Default Rproj values.
 path Location of the package

Examples

```
check_readme()
```

gitlab_instead_of *Gitlab rewrite*

Description

Rewrites the gitlab URL. This can be handy if you want to push to another GitLab repo on successful builds. Uses the environment variable CI_SERVER_HOST to determine the HOST.

Usage

```
gitlab_instead_of()
```

git_pre_commit *Adds a git hook*

Description

git_pre_commit only tests files that have changed. Makes things, especially linting a lot faster.

Add a pre-commit or pre-push hook. Both hooks are identical. They just get triggered at different times.

Usage

```
git_pre_commit()
```

```
add_pre_commit()
```

```
add_pre_push()
```

set_renvirion_var	<i>Sets an Renviron variable</i>
-------------------	----------------------------------

Description

Set an variable in the .Renviron file. Calls readEnviron to enable R to use the variable in future function calls.

Usage

```
set_renvirion_var(variable, value)
```

Arguments

variable	The variable name
value	The value

Index

`add_pre_commit` (`git_pre_commit`), 10

`add_pre_push` (`git_pre_commit`), 10

`build_pkg`, 2

`check_all`, 2

`check_all_news` (`check_news`), 5

`check_file_permissions`, 3

`check_gitignore`, 4

`check_lintr`, 4

`check_namespace`, 4

`check_news`, 5

`check_pkg`, 5

`check_r_filenames`, 6

`check_readme`, 6

`check_rproj` (`get_default_rproj`), 9

`check_tidy_description`, 7

`check_version`, 7

`check_via_env`, 8

`create_tag`, 9

`get_default_rproj`, 9

`get_pkg_tar_ball` (`build_pkg`), 2

`git_pre_commit`, 10

`gitlab_instead_of`, 10

`set_renvirion_var`, 11