Package ‘autonewsmd’

October 15, 2024

Title Auto-Generate Changelog using Conventional Commits

Version 0.0.9

Description Automatically generate a changelog file (NEWS.md / CHANGELOG.md) from the git history using conventional commit messages (<https://www.conventionalcommits.org/en/v1.0.0/>).

License GPL-3

URL https://github.com/kapsner/autonewsmd

BugReports https://github.com/kapsner/autonewsmd/issues

Depends R (>= 3.6)

Imports data.table, magrittr, quarto, R6, utils

Suggests git2r, lintr, testthat (>= 3.0.1)

VignetteBuilder quarto

Config/testthat/edition 3

Config/testthat/parallel false

Date/Publication 2024-10-15 11:00:02 UTC

Encoding UTF-8

SystemRequirements Quarto command line tools (https://github.com/quarto-dev/quarto-cli).

RoxygenNote 7.3.1

NeedsCompilation no

Author Lorenz A. Kapsner [cre, aut, cph] (<https://orcid.org/0000-0003-1866-860X>)

Maintainer Lorenz A. Kapsner <lorenz.kapsner@gmail.com>

Repository CRAN

Contents

`autonewsmd` *R6 Class to construct the changelog file*

Description

The `autonewsmd` class is used to construct a new changelog file, set all required metadata and to generate the file and save it to the root- directory of the respective repository.

Details

The changelog file has a title that typically includes the name of the repository, which needs at least to be provided when constructing a new changelog file.

Public fields

`repo_name` A character. The name of the repository, which is inserted into the title of the changelog file.

`tag_pattern` A character. A regular expression pattern to identify release tags in the repository. Defaults to `"^v(\d+\.){2}\d+(\.\d+)?$"` to identify patterns like `v0.0.1.9001`.

`repo_list` The list contains the commit messages prepared for the changelog file.

`file_name` A character. The name of the file, which is inserted into the title of the changelog file. Defaults to `"NEWS"` (typically one of `"NEWS"` or `"CHANGELOG"`).

`file_ending` A character. The file ending of the file the changelog should be written to. Defaults to `".md"`.

Methods

Public methods:

- `autonewsmd$new()`
- `autonewsmd$generate()`
- `autonewsmd$write()`
- `autonewsmd$clone()`

Method `new()`**:** Create a new `autonewsmd` object.

Usage:

```
autonewsmd$new(repo_name, repo_path = NULL, repo_remotes = NULL)
```

Arguments:

`repo_name` A character. The name of the repository, which is inserted into the title of the changelog file.

`repo_path` A character. The path of the repository to create a new changelog for. If `NULL` (the default), it will point automatically to to `"."`.

`repo_remotes` A character. The name of the tracked repository that should be used to get the repository's URL (e.g. when executing `git remote -v` in the shell). Defaults to `NULL`.

Returns: A new `autonewsmd` object.

Examples:

```
# (Example is based on the public examples from the `git2r` R package)
## Initialize a repository
path <- file.path(tempdir(), "autonewsmd")
dir.create(path)

if (requireNamespace("git2r", quietly = TRUE)) {
  repo <- git2r::init(path)

  ## Config user
  git2r::config(
    repo, user.name = "Alice", user.email = "alice@example.org"
  )
  git2r::remote_set_url(repo, "foobar", "https://example.org/git2r/foobar")


  ## Write to a file and commit
  lines <- paste0(
    "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do"
  )
  writeLines(lines, file.path(path, "example.txt"))

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "feat: new file")

  ## Write again to a file and commit
  lines2 <- paste0(
    "eiusmod tempor incididunt ut labore et dolore magna aliqua. ",
    "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris ",
    "nisi ut aliquip ex ea commodo consequat."
  )
  write(lines2, file.path(path, "example.txt"), append = TRUE)

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "refactor: added second phrase")

  ## now construct a new autonewsmd object
  an <- autonewsmd$new(repo_name = "TestRepo", repo_path = path)
}
```

Method `generate()`**:** Generate the list with the formatted commit messages that is used to render the changelog.

Usage:

```
autonewsmd$generate()
```

Details: The function generates the formatted list with the commit messages. If tags are available, each commit message is assigned to a specific tag. These assignments are used to structure the changelog document.

Returns: Populates the public field `repo_list`.

Examples:

```
# (Example is based on the public examples from the `git2r` R package)
## Initialize a repository
path <- file.path(tempdir(), "autonewsmd")
dir.create(path)

if (requireNamespace("git2r", quietly = TRUE)) {
  repo <- git2r::init(path)

  ## Config user
  git2r::config(
    repo, user.name = "Alice", user.email = "alice@example.org"
  )
  git2r::remote_set_url(repo, "foobar", "https://example.org/git2r/foobar")


  ## Write to a file and commit
  lines <- paste0(
    "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do"
  )
  writeLines(lines, file.path(path, "example.txt"))

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "feat: new file")

  ## Write again to a file and commit
  lines2 <- paste0(
    "eiusmod tempor incididunt ut labore et dolore magna aliqua. ",
    "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris ",
    "nisi ut aliquip ex ea commodo consequat."
  )
  write(lines2, file.path(path, "example.txt"), append = TRUE)

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "refactor: added second phrase")

  ## now construct a new autonewsmd object
  an <- autonewsmd$new(repo_name = "TestRepo", repo_path = path)

  ## generate the news and write them to the repo
  an$generate()
}
```

Method `write()`: Writes the changelog to the file system.

Usage:

```
autonewsmd$write(force = FALSE, con = NULL)
```

Arguments:

`force` A boolean. If `FALSE` (the default) a dialog is prompted to ask the user if the file should be (over-) written. If `TRUE`, the dialog is not prompted and the changelog file is created directly.

`con` A connection with the answer to the interactive question, if the changelog file should be written to the file system. This argument is intended mainly for being used in the unit tests.

Details: This function writes the changelog to the file system using the `file_name` and `file_ending` fields to compose the file name. CAUTION: existing files will be overwritten without any warning.

Returns: The function has no return value - it creates the new changelog file.

Examples:

```
# (Example is based on the public examples from the `git2r` R package)
## Initialize a repository
path <- tempdir()

if (requireNamespace("git2r", quietly = TRUE)) {
  repo <- git2r::init(path)

  ## Config user
  git2r::config(
    repo, user.name = "Alice", user.email = "alice@example.org"
  )
  git2r::remote_set_url(repo, "foobar", "https://example.org/git2r/foobar")


  ## Write to a file and commit
  lines <- paste0(
    "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do"
  )
  writeLines(lines, file.path(path, "example.txt"))

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "feat: new file")

  ## Write again to a file and commit
  lines2 <- paste0(
    "eiusmod tempor incididunt ut labore et dolore magna aliqua. ",
    "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris ",
    "nisi ut aliquip ex ea commodo consequat."
  )
  write(lines2, file.path(path, "example.txt"), append = TRUE)

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "refactor: added second phrase")

  ## now construct a new autonewsmd object
  an <- autonewsmd$new(repo_name = "TestRepo", repo_path = path)
```

```
    ## generate the news and write them to the repo
    an$generate()

    if (interactive()) {
      an$write()
    }
  }
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
autonewsmd$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
# (Example is based on the public examples from the `git2r` R package)
## Initialize a repository
path <- file.path(tempdir(), "autonewsmd")
dir.create(path)

if (requireNamespace("git2r", quietly = TRUE)) {
repo <- git2r::init(path)

  ## Config user
  git2r::config(repo, user.name = "Alice", user.email = "alice@example.org")
  git2r::remote_set_url(repo, "foobar", "https://example.org/git2r/foobar")


  ## Write to a file and commit
  lines <- "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do"
  writeLines(lines, file.path(path, "example.txt"))

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "feat: new file")

  ## Write again to a file and commit
  lines2 <- paste0(
    "eiusmod tempor incididunt ut labore et dolore magna aliqua. ",
    "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris ",
    "nisi ut aliquip ex ea commodo consequat."
  )
  write(lines2, file.path(path, "example.txt"), append = TRUE)

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "refactor: added second phrase")

  ## now construct a new autonewsmd object
  an <- autonewsmd$new(repo_name = "TestRepo", repo_path = path)
  ## generate the news and write them to the repo
  an$generate()

  if (interactive()) {
    an$write()
  }
}


## ------------------------------------------------
## Method `autonewsmd$new`
## ------------------------------------------------

# (Example is based on the public examples from the `git2r` R package)
## Initialize a repository
path <- file.path(tempdir(), "autonewsmd")
dir.create(path)

if (requireNamespace("git2r", quietly = TRUE)) {
  repo <- git2r::init(path)

  ## Config user
  git2r::config(
    repo, user.name = "Alice", user.email = "alice@example.org"
  )
  git2r::remote_set_url(repo, "foobar", "https://example.org/git2r/foobar")


  ## Write to a file and commit
  lines <- paste0(
    "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do"
  )
  writeLines(lines, file.path(path, "example.txt"))

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "feat: new file")

  ## Write again to a file and commit
  lines2 <- paste0(
    "eiusmod tempor incididunt ut labore et dolore magna aliqua. ",
    "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris ",
    "nisi ut aliquip ex ea commodo consequat."
  )
  write(lines2, file.path(path, "example.txt"), append = TRUE)

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "refactor: added second phrase")

  ## now construct a new autonewsmd object
  an <- autonewsmd$new(repo_name = "TestRepo", repo_path = path)
}
```

```
## ------------------------------------------------
## Method `autonewsmd$generate`
## ------------------------------------------------

# (Example is based on the public examples from the `git2r` R package)
## Initialize a repository
path <- file.path(tempdir(), "autonewsmd")
dir.create(path)

if (requireNamespace("git2r", quietly = TRUE)) {
  repo <- git2r::init(path)

  ## Config user
  git2r::config(
    repo, user.name = "Alice", user.email = "alice@example.org"
  )
  git2r::remote_set_url(repo, "foobar", "https://example.org/git2r/foobar")


  ## Write to a file and commit
  lines <- paste0(
    "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do"
  )
  writeLines(lines, file.path(path, "example.txt"))

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "feat: new file")

  ## Write again to a file and commit
  lines2 <- paste0(
    "eiusmod tempor incididunt ut labore et dolore magna aliqua. ",
    "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris ",
    "nisi ut aliquip ex ea commodo consequat."
  )
  write(lines2, file.path(path, "example.txt"), append = TRUE)

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "refactor: added second phrase")

  ## now construct a new autonewsmd object
  an <- autonewsmd$new(repo_name = "TestRepo", repo_path = path)

  ## generate the news and write them to the repo
  an$generate()
}


## ------------------------------------------------
## Method `autonewsmd$write`
## ------------------------------------------------

# (Example is based on the public examples from the `git2r` R package)
## Initialize a repository
```

```
path <- tempdir()

if (requireNamespace("git2r", quietly = TRUE)) {
  repo <- git2r::init(path)

  ## Config user
  git2r::config(
    repo, user.name = "Alice", user.email = "alice@example.org"
  )
  git2r::remote_set_url(repo, "foobar", "https://example.org/git2r/foobar")


  ## Write to a file and commit
  lines <- paste0(
    "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do"
  )
  writeLines(lines, file.path(path, "example.txt"))

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "feat: new file")

  ## Write again to a file and commit
  lines2 <- paste0(
    "eiusmod tempor incididunt ut labore et dolore magna aliqua. ",
    "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris ",
    "nisi ut aliquip ex ea commodo consequat."
  )
  write(lines2, file.path(path, "example.txt"), append = TRUE)

  git2r::add(repo, "example.txt")
  git2r::commit(repo, "refactor: added second phrase")

  ## now construct a new autonewsmd object
  an <- autonewsmd$new(repo_name = "TestRepo", repo_path = path)

  ## generate the news and write them to the repo
  an$generate()

  if (interactive()) {
    an$write()
  }
}
```

Index