

# Package ‘allcontributors’

May 7, 2026

**Title** Acknowledge all Contributors to a Project

**Version** 0.2.3

**Description** Acknowledge all contributors to a project via a single function call. The function appends to a 'README' or other specified file(s) a table with names of all individuals who contributed via code or repository issues. The package also includes several additional functions to extract and quantify contributions to any repository.

**License** GPL-3

**URL** <https://github.com/ropensci/allcontributors>

**BugReports** <https://github.com/ropensci/allcontributors/issues>

**Imports** cli, clipr, curl, gert, gh, gitcreds, magrittr

**Suggests** httptest2, knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-GB

**RoxygenNote** 7.3.3

**Config/ropensci/maintainer** staff

**NeedsCompilation** no

**Author** Mark Padgham [aut, cre],  
Chris Hartgerink [aut],  
Mælle Salmon [ctb]

**Maintainer** Mark Padgham <mark.padgham@email.com>

**Repository** CRAN

**Date/Publication** 2026-04-02 10:50:07 UTC

## Contents

add_contributors . . . . .	2
get_contributors . . . . .	4

get_gh_code_contributors . . . . .	5
get_gh_contrib_issue . . . . .	6
get_gh_issue_people . . . . .	7
get_gh_issue_titles . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

add_contributors	<i>add_contributors</i>
------------------	-------------------------

---

## Description

Add contributors to README.Rmd

## Usage

```
add_contributors(
  repo = ".",
  ncols = 7,
  files = c("README.Rmd", "README.md"),
  type = c("code", "issues", "discussion"),
  exclude_label = "wontfix",
  exclude_issues = NULL,
  exclude_not_planned = TRUE,
  num_sections = 3,
  section_names = c("Code", "Issue Authors", "Issue Contributors"),
  format = "grid",
  check_urls = TRUE,
  alphabetical = FALSE,
  open_issue = FALSE,
  force_update = FALSE
)
```

## Arguments

repo	Vector of repository locations for which contributions are to be extracted. Each location must be a git project with a github remote. Default is single repository in current working directory.
ncols	Number of columns for contributors in 'README'
files	Names of files in which to add contributors
type	Type of contributions to include: 'code' for direct code contributions (including documentation), 'issues' to recognise contributors who open issues, and 'discussion' for contributing to discussions within issues. Discussion contributions are only from individuals not present in either 'issues' or 'code'; and 'issues' contributions are only from individuals not present in 'code'.
exclude_label	Exclude any contributions from issues with specified label (default = "wontfix"; set to NULL or empty string to include all issues).

exclude_issues	Numbers of any issues (or pull requests) to be excluded from lists of contributors.
exclude_not_planned	If TRUE (default), exclude contributions to any issues closed as "not planned".
num_sections	Number of sections in which to divide contributors: <ul style="list-style-type: none"><li>• 1 All contributions within single section regardless of type</li><li>• 2 Contributions divided between a single section for code and a second section for all other issue-related contributions.</li><li>• 3 Contributions divided into single sections for each of the three type arguments.</li></ul>
section_names	Names of the sections to appear on the nominated files.
format	One of ("grid", "list", "text") to control display of contributors as <ul style="list-style-type: none"><li>• 1 "grid" for a rectangular grid, with each contributor represented by their github avatar, with avatar linked to contributor profile and name linked to repository contributions.</li><li>• 2 "list" for a more condensed list with github user names only and no avatars, one contributor per line linked to issue contributions.</li><li>• 3 "text" for a single line of text containing comma-separated github user names linked to issue contributions.</li></ul>
check_urls	If TRUE (default), GitHub URLs of all contributors are checked to ensure they are still valid. (This is generally the most time-consuming stage, so set to 'FALSE' if you are sure all URLs are valid.)
alphabetical	If TRUE, order contributors alphabetically, otherwise order by decreasing numbers of contributions.
open_issue	If TRUE, open or edit an issue on github in order to notify all contributors that they've been added to your README (see Note).
force_update	If TRUE, update the specified files even if contributions have not changed.

**Value**

Named list of logical values indicating whether files of given names were updated or not is returned invisibly (that is, only if explicitly assigned to a return value).

**Note**

Opening an issue on github requires the github command-line interface to be locally installed. See <https://cli.github.com/>.

**See Also**

Other main: [get\\_contributors\(\)](#)

**Examples**

```
# The following code extracts the contributors from the git repository
# associated with current working directory and writes them to a file.
## Not run:
f <- tempfile (fileext = ".Rmd")
writeLines ("", f) # blank file in tempdir()
add_contributors (repo = ".", files = f)

## End(Not run)
```

---

```
get_contributors      get_contributors
```

---

**Description**

Get all contributors to a repository, including those who contribute to code, open issues, and contribute to discussions in issues.

**Usage**

```
get_contributors(
  org,
  repo,
  type = c("code", "issues", "discussion"),
  exclude_label = "wontfix",
  exclude_issues = NULL,
  exclude_not_planned = TRUE,
  alphabetical = FALSE,
  rm_ctbs_ptn = "\\-(bot|actions)$",
  check_urls = TRUE,
  quiet = FALSE
)
```

**Arguments**

org	Github organisation name for repository
repo	Repository within org for which contributors are to be extracted
type	Type of contributions to include: 'code' for direct code contributions (including documentation), 'issues' to recognise contributors who open issues, and 'discussion' for contributing to discussions within issues. Discussion contributions are only from individuals not present in either 'issues' or 'code'; and 'issues' contributions are only from individuals not present in 'code'.
exclude_label	Exclude any contributions from issues with specified label (default = "wontfix"; set to NULL or empty string to include all issues).
exclude_issues	Numbers of any issues (or pull requests) to be excluded from lists of contributors.

exclude_not_planned	If TRUE (default), exclude contributions to any issues closed as "not planned".
alphabetical	If TRUE, order contributors alphabetically, otherwise order by decreasing numbers of contributions.
rm_ctbs_ptn	Remove contributors which match this pattern (such as "actions-bot", or "ropensci-review-bot").
check_urls	If TRUE (default), GitHub URLs of all contributors are checked to ensure they are still valid. (This is generally the most time-consuming stage, so set to 'FALSE' if you are sure all URLs are valid.)
quiet	If FALSE, display progress information on screen.

**See Also**

Other main: [add\\_contributors\(\)](#)

**Examples**

```
## Not run:
get_contributors (org = "ropensci", repo = "allcontributors")

## End(Not run)
```

---

```
get_gh_code_contributors
      get_gh_code_contributors
```

---

**Description**

Get list of all code contributors to the code of a repository

**Usage**

```
get_gh_code_contributors(org, repo, alphabetical = FALSE)
```

**Arguments**

org	Github organisation name for repository
repo	Repository within org for which contributors are to be extracted
alphabetical	If TRUE, order contributors alphabetically, otherwise order by decreasing numbers of contributions.

**Value**

A data.frame of two columns of contributor (name, login)

**See Also**

Other github: [get\\_gh\\_contrib\\_issue\(\)](#), [get\\_gh\\_issue\\_people\(\)](#), [get\\_gh\\_issue\\_titles\(\)](#)

**Examples**

```
## Not run:  
get_gh_code_contributors (org = "ropensci", repo = "allcontributors")  
  
## End(Not run)
```

---

`get_gh_contrib_issue` *get\_gh\_contrib\_issue*

---

**Description**

Extract contributors currently listed on an "All Contributions" issue in a github repository.

**Usage**

```
get_gh_contrib_issue(org, repo)
```

**Arguments**

<code>org</code>	Github organisation name for repository
<code>repo</code>	Repository within org for which contributors are to be extracted

**Value**

Character vector of github logins for all contributors listed in current issue, or empty character string if there no issue named "All Contributors".

**See Also**

Other github: [get\\_gh\\_code\\_contributors\(\)](#), [get\\_gh\\_issue\\_people\(\)](#), [get\\_gh\\_issue\\_titles\(\)](#)

**Examples**

```
## Not run:  
get_gh_contrib_issue (org = "ropensci", repo = "allcontributors")  
  
## End(Not run)
```

---

get\_gh\_issue\_people    *get\_gh\_issue\_people*

---

### Description

Extract lists of (1) all authors of, and (2) all contributors to, all github issues for nominated repository, excluding issues closed as "not planned"

### Usage

```
get_gh_issue_people(  
  org,  
  repo,  
  exclude_issues = NULL,  
  exclude_label = "wontfix",  
  exclude_not_planned = TRUE  
)
```

### Arguments

org	Github organisation name for repository
repo	Repository within org for which contributors are to be extracted
exclude_issues	Numbers of any issues (or pull requests) to be excluded from lists of contributors.
exclude_label	Exclude any contributions from issues with specified label (default = "wontfix"; set to NULL or empty string to include all issues).
exclude_not_planned	If TRUE (default), exclude contributions to any issues closed as "not planned".

### Value

List of (authors, contributors), each as character vector of github login names.

### See Also

Other github: [get\\_gh\\_code\\_contributors\(\)](#), [get\\_gh\\_contrib\\_issue\(\)](#), [get\\_gh\\_issue\\_titles\(\)](#)

### Examples

```
## Not run:  
get_gh_issue_people (org = "ropensci", repo = "allcontributors")  
  
## End(Not run)
```

---

`get_gh_issue_titles`    *get\_gh\_issue\_titles*

---

**Description**

Extract titles and numbers of all issues associated with a nominated repository

**Usage**

```
get_gh_issue_titles(org, repo)
```

**Arguments**

<code>org</code>	Github organisation name for repository
<code>repo</code>	Repository within org for which contributors are to be extracted

**Value**

data.frame with one column of issue numbers, and one column of issue titles.

**See Also**

Other github: [get\\_gh\\_code\\_contributors\(\)](#), [get\\_gh\\_contrib\\_issue\(\)](#), [get\\_gh\\_issue\\_people\(\)](#)

**Examples**

```
## Not run:  
get_gh_issue_titles (org = "ropensci", repo = "allcontributors")  
  
## End(Not run)
```

# Index

## \* **github**

- get\_gh\_code\_contributors, 5
- get\_gh\_contrib\_issue, 6
- get\_gh\_issue\_people, 7
- get\_gh\_issue\_titles, 8

## \* **main**

- add\_contributors, 2
- get\_contributors, 4

add\_contributors, 2, 5

get\_contributors, 3, 4

get\_gh\_code\_contributors, 5, 6–8

get\_gh\_contrib\_issue, 6, 6, 7, 8

get\_gh\_issue\_people, 6, 7, 8

get\_gh\_issue\_titles, 6, 7, 8