

Package: roreviewapi (via r-universe)

October 23, 2026

Title Plumber API to report package structure and function

Version 0.2.1.051

Description Plumber API to report package structure and function.

License GPL-3

URL <https://docs.ropensci.org/roreviewapi>,
<https://github.com/ropensci-review-tools/roreviewapi>

BugReports <https://github.com/ropensci-review-tools/roreviewapi/issues>

Depends R (>= 4.1.0)

Imports airtabler, BiocManager, callr, commonmark, DBI, fs, gert, gh,
gmailr, here, htr2, jsonlite, later, logger, magrittr,
methods, openssl, pak, pkgcheck, plumber, rappdirs, RSQLite,
srr, tictoc, xml2

Suggests bspm, knitr, pkgmatch, rmarkdown, roxygen2, rvest, testthat
(>= 3.0.0), withr

VignetteBuilder knitr

Remotes bergant/airtabler, ropensci-review-tools/pkgcheck,
ropensci-review-tools/pkgmatch

Encoding UTF-8

Language en-GB

LazyData true

NeedsCompilation yes

Roxygen list(markdown = TRUE)

SystemRequirements gh, dos2unix

Config/testthat/edition 3

Config/ropensci/maintainer staff

Config/roxygen2/version 8.1.1

Config/pak/sysreqs cmake git libglpk-dev make libgit2-dev libicu-dev libsodium-dev libuv1-
dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-10-23 06:12:42 UTC

RemoteUrl <https://github.com/ropensci-review-tools/roreviewapi>

RemoteRef main

RemoteSha e081f788c238a7bcb848b864d3a55a3c20fd63c5

Contents

check_cache	3
check_issue_template	3
collate_editor_check	4
cran_to_list	5
deactivate_search	5
deactivate_stale_searches	6
dl_gh_repo	6
editor_check	7
get_branch_from_url	7
get_subdir_from_url	8
handle_click	8
is_user_authorized	9
issue_is_stats	10
list_searches	10
pkgmatch_repo	11
pkgrep_install_deps	11
post_to_issue	12
push_to_gh_pages	13
readme_badge	13
readme_has_peer_review_badge	14
ros_to_list	14
send_search	15
send_search_and_post	16
serve_api	17
srr_counts	17
srr_counts_from_report	18
srr_counts_summary	19
stats_badge	19
stdout_stderr_cache	20
symbol_crs	20
symbol_tck	21
url_is_r_pkg	21

Index	22
--------------	-----------

check_cache	<i>check_cache</i>
-------------	--------------------

Description

Check whether a package has been cached, and if so, whether commits have been added to the github repo since cached version.

Usage

```
check_cache(org, repo, cache_dir = fs::path_temp())
```

Arguments

org	Github organization
repo	Github repository
cache_dir	Directory in which packages are cached

Value

FALSE If a package has previously been cached, and the github repo has not changed; otherwise TRUE.

Note

This function is not intended to be called directly, and is only exported to enable it to be used within the **plumber** API.

See Also

Other utils: [pkgrep_install_deps\(\)](#), [stdout_stderr_cache\(\)](#), [symbol_crs\(\)](#), [symbol_tck\(\)](#)

check_issue_template	<i>Check template variables in GitHub issue</i>
----------------------	---

Description

Check template variables in GitHub issue

Usage

```
check_issue_template(orgrepo, issue_num)
```

Arguments

orgrepo	GitHub organization and repo as single string separated by forward slash (org/repo).
issue_num	Number of issue from which to extract opening comment

Value

Comment as character string

See Also

Other ropensci: [is_user_authorized\(\)](#), [issue_is_stats\(\)](#), [push_to_gh_pages\(\)](#), [readme_has_peer_review_badge\(\)](#), [srr_counts\(\)](#), [srr_counts_from_report\(\)](#), [srr_counts_summary\(\)](#), [stats_badge\(\)](#)

collate_editor_check	<i>Collate list of checks to single concatenated character string</i>
----------------------	---

Description

Collate list of checks to single concatenated character string

Usage

```
collate_editor_check(checks)
```

Arguments

checks	Result of <code>pkgcheck::pkgcheck</code> function
--------	--

Value

Single character

Note

Exported only for access by plumber; not intended for general external usage.

See Also

Other main: [editor_check\(\)](#), [serve_api\(\)](#)

cran_to_list	<i>Convert names of CRAN packages to markdown-formatted list.</i>
--------------	---

Description

This function is exported because it needs to be called in the main plumber endpoint function.

Usage

```
cran_to_list(matches, n = 5L)
```

Arguments

matches	A pkgmatch <code>data.frame</code> object with columns of ("package", "version", "rank").
n	Number of matches to return in list.

See Also

Other pkgmatch: [pkgmatch_repo\(\)](#), [ros_to_list\(\)](#)

deactivate_search	<i>Deactivate a volunteer search and delete all associated data</i>
-------------------	---

Description

Deactivate a volunteer search and delete all associated data

Usage

```
deactivate_search(repo, issue_id)
```

Arguments

repo	GitHub review repository in org/repo format.
issue_id	Integer issue number in the review repository.

Value

Named list with deactivated (logical) and issue_ref.

See Also

Other email: [deactivate_stale_searches\(\)](#), [handle_click\(\)](#), [list_searches\(\)](#), [send_search\(\)](#), [send_search_and_post\(\)](#)

deactivate_stale_searches

Auto-deactivate and delete stale volunteer searches

Description

Searches are never automatically closed otherwise, so this is intended to be called periodically (see `serve_api()`) to stop searches that are never explicitly deactivated from accumulating in the database forever. Any search whose `created_at` is older than `max_age_days` is deactivated and has all its data removed, exactly as `deactivate_search()` does for a single search.

Usage

```
deactivate_stale_searches(max_age_days = 100L)
```

Arguments

`max_age_days` Default: 100. Integer number of days after creation at which a search is considered stale and is automatically deactivated.

Value

Integer count of searches deactivated, invisibly.

See Also

Other email: [deactivate_search\(\)](#), [handle_click\(\)](#), [list_searches\(\)](#), [send_search\(\)](#), [send_search_and_post\(\)](#)

dl_gh_repo

Download a GitHub repo to local cache

Description

Download a GitHub repo to local cache

Usage

```
dl_gh_repo(u, branch = NULL)
```

Arguments

`u` URL of GitHub repository
`branch` Checkout specified (non-default) branch of repo.

Value

Path to locally cached '.zip' version of repository

See Also

Other github: [get_branch_from_url\(\)](#), [get_subdir_from_url\(\)](#), [post_to_issue\(\)](#), [url_is_r_pkg\(\)](#)

editor_check	<i>Body of main 'editorcheck' response</i>
--------------	--

Description

Body of main 'editorcheck' response

Usage

```
editor_check(repourl, repo, issue_id, post_to_issue = TRUE)
```

Arguments

repourl	The URL for the repo being checked, potentially including full path to non-default branch.
repo	The 'context.repo' parameter defining the repository from which the command was invoked, passed in 'org/repo' format.
issue_id	The id (number) of the issue from which the command was invoked.
post_to_issue	Integer value > 0 will post results back to issue (via 'gh' cli); otherwise just return character string with result.

Value

If !post_to_issue, a markdown-formatted response body from static package checks, otherwise URL of the issue comment to which response body has been posted.

See Also

Other main: [collate_editor_check\(\)](#), [serve_api\(\)](#)

get_branch_from_url	<i>Get branch from a GitHub URL if non-default branch specified there</i>
---------------------	---

Description

Get branch from a GitHub URL if non-default branch specified there

Usage

```
get_branch_from_url(repourl)
```

Arguments

repourl Potentially with "/tree/branch_name/sub-directory" appended

Value

Branch as single string.

See Also

Other github: [dl_gh_repo\(\)](#), [get_subdir_from_url\(\)](#), [post_to_issue\(\)](#), [url_is_r_pkg\(\)](#)

get_subdir_from_url	<i>Return sub-directory from URL if present. This function is also intended to test whether packages are in sub-directories (issue #64)</i>
---------------------	---

Description

Return sub-directory from URL if present. This function is also intended to test whether packages are in sub-directories (issue #64)

Usage

```
get_subdir_from_url(repourl)
```

Arguments

repourl Potentially with "/tree/branch_name/sub-directory" appended

See Also

Other github: [dl_gh_repo\(\)](#), [get_branch_from_url\(\)](#), [post_to_issue\(\)](#), [url_is_r_pkg\(\)](#)

handle_click	<i>Handle a volunteer link click</i>
--------------	--------------------------------------

Description

Looks up the token, checks whether the parent search is still active, guards against double-clicks, records the click timestamp, and triggers a Gmail API notification (Phase 2).

Usage

```
handle_click(token, sender = NULL)
```

Arguments

token	64-character hex token from the recipient's unique link.
sender	Function used to send the click notification; injectable for testing. Must accept (to, subject, html_body). Defaults to gmail_send.

Value

Named list with status (integer HTTP status code) and body (character HTML string).

See Also

Other email: [deactivate_search\(\)](#), [deactivate_stale_searches\(\)](#), [list_searches\(\)](#), [send_search\(\)](#), [send_search_and_post\(\)](#)

is_user_authorized	<i>Check whether a caller is authorized to call endpoints.</i>
--------------------	--

Description

This function is used only in the **plumber** endpoints, to prevent them being called by unauthorized users. Authorization is determined solely from the secret supplied by the caller, and never from the token hard-coded in the container. The secret is authorized if it either matches the environment variable PKGCHECK_TOKEN, or is a GitHub token belonging to one of the authorized users.

Usage

```
is_user_authorized(secret = NULL)
```

Arguments

secret	Environment variable PKGCHECK_TOKEN sent from bot, or a GitHub token of an authorized user.
--------	---

Value

Logical value indicating whether or not a user is authorized.

See Also

Other ropensci: [check_issue_template\(\)](#), [issue_is_stats\(\)](#), [push_to_gh_pages\(\)](#), [readme_has_peer_review_badge\(\)](#), [srr_counts\(\)](#), [srr_counts_from_report\(\)](#), [srr_counts_summary\(\)](#), [stats_badge\(\)](#)

issue_is_stats	<i>Determine whether a GitHub issue is a Stats submission</i>
----------------	---

Description

Determine whether a GitHub issue is a Stats submission

Usage

```
issue_is_stats(orgrepo, issue_num)
```

Arguments

orgrepo	GitHub organization and repo as single string separated by forward slash (org/repo).
issue_num	Number of issue from which to extract submission type.

Value

TRUE if the submission type is "Stats", otherwise FALSE.

See Also

Other ropensci: [check_issue_template\(\)](#), [is_user_authorized\(\)](#), [push_to_gh_pages\(\)](#), [readme_has_peer_review_b](#), [srr_counts\(\)](#), [srr_counts_from_report\(\)](#), [srr_counts_summary\(\)](#), [stats_badge\(\)](#)

list_searches	<i>List all volunteer searches with recipient and click counts</i>
---------------	--

Description

List all volunteer searches with recipient and click counts

Usage

```
list_searches()
```

Value

Data frame with columns search_id, created_at, notify_email, active, total, clicked.

See Also

Other email: [deactivate_search\(\)](#), [deactivate_stale_searches\(\)](#), [handle_click\(\)](#), [send_search\(\)](#), [send_search_and_post\(\)](#)

pkgmatch_repo	<i>Body of main 'pkgmatch' response</i>
---------------	---

Description

Body of main 'pkgmatch' response

Usage

```
pkgmatch_repo(repourl, repo, issue_id, n_top = 5L, post_to_issue = TRUE)
```

Arguments

repourl	The URL for the repo being matched, potentially including full path to non-default branch.
repo	The 'context.repo' parameter defining the repository from which the command was invoked, passed in 'org/repo' format.
issue_id	The id (number) of the issue from which the command was invoked.
n_top	Return this number of top-matches packages.
post_to_issue	Integer value > 0 will post results back to issue (via 'gh' cli); otherwise just return character string with result.

Value

If !post_to_issue, a markdown-formatted response body, otherwise URL of the issue comment to which response body has been posted.

See Also

Other pkgmatch: [cran_to_list\(\)](#), [ros_to_list\(\)](#)

pkgrep_install_deps	<i>Install all system and package dependencies of an R package</i>
---------------------	--

Description

Install all system and package dependencies of an R package

Usage

```
pkgrep_install_deps(path, repo, issue_id)
```

Arguments

path	Path to local file or directory
repo	The 'context.repo' parameter defining the repository from which the command was invoked, passed in 'org/repo' format.
issue_id	The id (number) of the issue from which the command was invoked.

Value

Hopefully a character vector of length zero, otherwise a message detailing any R packages unable to be installed.

See Also

Other utils: [check_cache\(\)](#), [stdout_stderr_cache\(\)](#), [symbol_crs\(\)](#), [symbol_tck\(\)](#)

post_to_issue	<i>Post review checks to GitHub issue</i>
---------------	---

Description

Post review checks to GitHub issue

Usage

```
post_to_issue(cmt, repo, issue_id)
```

Arguments

cmt	Single character string with comment to post.
repo	The repository to post to (obtained directly from bot).
issue_id	The number of the issue to post to.

Value

URL of the comment within the nominated issue

See Also

Other github: [dl_gh_repo\(\)](#), [get_branch_from_url\(\)](#), [get_subdir_from_url\(\)](#), [url_is_r_pkg\(\)](#)

push_to_gh_pages	<i>Push static html files to gh-pages branch of this repo to serve via GitHub pages.</i>
------------------	--

Description

Push static html files to gh-pages branch of this repo to serve via GitHub pages.

Usage

```
push_to_gh_pages(check)
```

Arguments

check	Return result of editor_check function.
-------	---

Value

Vector of two paths containing URLs of the srr and network files.

See Also

Other ropensci: [check_issue_template\(\)](#), [is_user_authorized\(\)](#), [issue_is_stats\(\)](#), [readme_has_peer_review_badge\(\)](#), [srr_counts\(\)](#), [srr_counts_from_report\(\)](#), [srr_counts_summary\(\)](#), [stats_badge\(\)](#)

readme_badge	<i>Check whether README.md features an rOpenSci software-review badge</i>
--------------	---

Description

Check whether README.md features an rOpenSci software-review badge

Usage

```
readme_badge(repourl, repo = NULL, issue_id = NULL, post_to_issue = TRUE)
```

Arguments

repourl	The URL for the repo being checked, potentially including full path to non-default branch.
repo	The 'context.repo' parameter defining the repository from which the command was invoked, passed in 'org/repo' format.
issue_id	The id (number) of the issue from which the command was invoked.
post_to_issue	Integer value > 0 will post results back to issue (via 'gh' cli); otherwise just return character string with result.

Value

A string, empty if the badge was found.

```
readme_has_peer_review_badge
```

Check whether 'README.md' has a "peer reviewed" badge

Description

Check whether 'README.md' has a "peer reviewed" badge

Usage

```
readme_has_peer_review_badge(path = getwd(), issue_id = NULL)
```

Arguments

path	Local path to package directory.
issue_id	The id (number) of the issue from which the command was invoked.

Value

A string, empty if the badge was found.

See Also

Other ropensci: [check_issue_template\(\)](#), [is_user_authorized\(\)](#), [issue_is_stats\(\)](#), [push_to_gh_pages\(\)](#), [srr_counts\(\)](#), [srr_counts_from_report\(\)](#), [srr_counts_summary\(\)](#), [stats_badge\(\)](#)

```
ros_to_list
```

Convert names of rOpenSci packages to markdown-formatted list.

Description

This function is exported because it needs to be called in the main plumber endpoint function.

Usage

```
ros_to_list(matches, n = 5L)
```

Arguments

matches	A pkgmatch data.frame object with columns of ("package", "version", "rank").
n	Number of matches to return in list.

See Also

Other pkgmatch: [cran_to_list\(\)](#), [pkgmatch_repo\(\)](#)

send_search	<i>Send a batch of editor search emails</i>
-------------	---

Description

Fetches current editor email addresses via `get_editor_emails()`, inserts a new search record and one recipient row per address into the database, then dispatches emails via the Gmail API. The notify address is read from the AirTable cache written by the internal `'notify_email_refresh'` function. The base URL for click links is read from the `ROREVIEWAPI_BASE_URL` environment variable. The stats/standard distinction is determined by calling `issue_is_stats()` on the supplied repo and `issue_id`.

Usage

```
send_search(
  repourl,
  repo,
  issue_id,
  subject = "Seeking editors for rOpenSci software submission",
  fetcher = NULL,
  stats_checker = roreviewapi::issue_is_stats,
  sender = NULL
)
```

Arguments

<code>repourl</code>	URL of the package repository; included in the outgoing emails.
<code>repo</code>	GitHub review repository in <code>org/repo</code> format.
<code>issue_id</code>	Integer issue number in the review repository.
<code>subject</code>	Subject line for the outgoing emails.
<code>fetcher</code>	Function used to fetch editor emails; injectable for testing. Must accept (<code>airtable_base_id</code> , <code>stats</code>) and return a character vector.
<code>stats_checker</code>	Function used to determine submission type; injectable for testing. Must accept (<code>repo</code> , <code>issue_id</code>) and return a logical.
<code>sender</code>	Function used to send the batch of emails; injectable for testing. Must accept (<code>emails</code> , <code>links</code> , <code>subject</code> , <code>repo</code> , <code>issue_id</code>) and return a list of per-recipient responses. Defaults to <code>gmail_send_batch</code> .

Value

Named list with `search_id` (integer) and `sent` (integer).

See Also

Other email: [deactivate_search\(\)](#), [deactivate_stale_searches\(\)](#), [handle_click\(\)](#), [list_searches\(\)](#), [send_search_and_post\(\)](#)

send_search_and_post *Send a batch of editor search emails, and post the outcome to the issue*

Description

Wrapper around `send_search()` intended to be run as a background process from the `send_search` endpoint, which returns before any emails are sent. The outcome, success or failure, is posted back to the GitHub issue from which the command was invoked.

Usage

```
send_search_and_post(
    repourl,
    repo,
    issue_id,
    poster = roreviewapi::post_to_issue,
    ...
)
```

Arguments

<code>repourl</code>	URL of the package repository; included in the outgoing emails.
<code>repo</code>	GitHub review repository in <code>org/repo</code> format.
<code>issue_id</code>	Integer issue number in the review repository.
<code>poster</code>	Function used to post the outcome to the issue; injectable for testing. Must accept <code>(cmt, repo, issue_id)</code> . Defaults to <code>post_to_issue</code> .
<code>...</code>	Further arguments passed to <code>send_search()</code> .

Value

The comment posted to the issue, invisibly.

See Also

Other email: [deactivate_search\(\)](#), [deactivate_stale_searches\(\)](#), [handle_click\(\)](#), [list_searches\(\)](#), [send_search\(\)](#)

serve_api	<i>serve plumber API to report on packages</i>
-----------	--

Description

The API exposes the single POST points of report to download software from the given URL and return a textual analysis of its structure and functionality.

Usage

```
serve_api(port = 8000L, cache_dir = NULL, os = "", os_release = "")
```

Arguments

port	Port for API to be exposed on
cache_dir	Directory where previously downloaded repositories are cached
os	Name of operating system, passed to remotes package to install system dependencies.
os_release	Name of operating system release, passed to remotes package to install system dependencies.

Value

Nothing; calling this starts a blocking process.

See Also

Other main: [collate_editor_check\(\)](#), [editor_check\(\)](#)

srr_counts	<i>Count number of 'srr' statistical standards complied with, and confirm whether than represents > 50% of all applicable standards.</i>
------------	---

Description

Count number of 'srr' statistical standards complied with, and confirm whether than represents > 50% of all applicable standards.

Usage

```
srr_counts(repourl, repo, issue_id, post_to_issue = TRUE)
```

Arguments

repourl	The URL for the repo being checked, potentially including full path to non-default branch.
repo	The 'context.repo' parameter defining the repository from which the command was invoked, passed in 'org/repo' format.
issue_id	The id (number) of the issue from which the command was invoked.
post_to_issue	Integer value > 0 will post results back to issue (via 'gh' cli); otherwise just return character string with result.

Value

Vector of three numbers:

1. Number of standards complied with
2. Total number of applicable standards
3. Number complied with as proportion of total

See Also

Other ropensci: [check_issue_template\(\)](#), [is_user_authorized\(\)](#), [issue_is_stats\(\)](#), [push_to_gh_pages\(\)](#), [readme_has_peer_review_badge\(\)](#), [srr_counts_from_report\(\)](#), [srr_counts_summary\(\)](#), [stats_badge\(\)](#)

srr_counts_from_report

Extract final counts of 'srr' standards from the report

Description

Extract final counts of 'srr' standards from the report

Usage

```
srr_counts_from_report(srr_rep)
```

Arguments

srr_rep An 'srr' report generated by the `srr::srr_report()` function.

Value

Character vector with markdown-formatted summary summary of numbers of standards complied with.

See Also

Other ropensci: [check_issue_template\(\)](#), [is_user_authorized\(\)](#), [issue_is_stats\(\)](#), [push_to_gh_pages\(\)](#), [readme_has_peer_review_badge\(\)](#), [srr_counts\(\)](#), [srr_counts_summary\(\)](#), [stats_badge\(\)](#)

srr_counts_summary	<i>Summarise counts of 'srr' standards from full 'srr' report</i>
--------------------	---

Description

Summarise counts of 'srr' standards from full 'srr' report

Usage

```
srr_counts_summary(srr_rep, has_errors = FALSE)
```

Arguments

srr_rep	An 'srr' report generated by the <code>srr::srr_report()</code> function.
has_errors	This is TRUE when the 'srr' report has errors, in which case the final message issued here excludes statements about package being able to be submitted.

Value

Character vector with markdown-formatted summary summary of numbers of standards complied with.

See Also

Other ropensci: [check_issue_template\(\)](#), [is_user_authorized\(\)](#), [issue_is_stats\(\)](#), [push_to_gh_pages\(\)](#), [readme_has_peer_review_badge\(\)](#), [srr_counts\(\)](#), [srr_counts_from_report\(\)](#), [stats_badge\(\)](#)

stats_badge	<i>Get stats badge grade and standards version for a submission</i>
-------------	---

Description

Get stats badge grade and standards version for a submission

Usage

```
stats_badge(repo = "ropensci/software-review", issue_num = 258)
```

Arguments

repo	The submission repo
issue_num	GitHub issue number of submission

Value

A single character containing the label used directly for the issue badge

See Also

Other ropensci: [check_issue_template\(\)](#), [is_user_authorized\(\)](#), [issue_is_stats\(\)](#), [push_to_gh_pages\(\)](#), [readme_has_peer_review_badge\(\)](#), [srr_counts\(\)](#), [srr_counts_from_report\(\)](#), [srr_counts_summary\(\)](#)

stdout_stderr_cache	<i>Set up stdout & stderr cache files for r_bg process</i>
---------------------	--

Description

Set up stdout & stderr cache files for r_bg process

Usage

```
stdout_stderr_cache(repourl)
```

Arguments

repourl	The URL of the repo being checked
---------	-----------------------------------

Value

Vector of two strings holding respective local paths to stdout and stderr files for r_bg process controlling the main [editor_check](#) function.

Note

These files are needed for the **callr** r_bg process which controls the main [editor_check](#) call. See <https://github.com/r-lib/callr/issues/204>. The stdout and stderr pipes from the process are stored in the cache directory so they can be inspected via their own distinct endpoint calls.

See Also

Other utils: [check_cache\(\)](#), [pkgrep_install_deps\(\)](#), [symbol_crs\(\)](#), [symbol_tck\(\)](#)

symbol_crs	<i>Cross symbol, exported for direct use in plumber API</i>
------------	---

Description

Cross symbol, exported for direct use in plumber API

Usage

```
symbol_crs()
```

See Also

Other utils: [check_cache\(\)](#), [pkgrep_install_deps\(\)](#), [stdout_stderr_cache\(\)](#), [symbol_tck\(\)](#)

symbol_tck	<i>Tick symbol, exported for direct use in plumber API</i>
------------	--

Description

Tick symbol, exported for direct use in plumber API

Usage

```
symbol_tck()
```

See Also

Other utils: [check_cache\(\)](#), [pkgrep_install_deps\(\)](#), [stdout_stderr_cache\(\)](#), [symbol_crs\(\)](#)

url_is_r_pkg	<i>Check whether a given GitHub URL contains an R package.</i>
--------------	--

Description

Check whether a given GitHub URL contains an R package.

Usage

```
url_is_r_pkg(repourl)
```

Arguments

repourl	Potentially with "/tree/branch_name/sub-directory" appended
---------	---

Value

TRUE if 'repourl' is an R package, otherwise FALSE

See Also

Other github: [dl_gh_repo\(\)](#), [get_branch_from_url\(\)](#), [get_subdir_from_url\(\)](#), [post_to_issue\(\)](#)

Index

- * **email**
 - deactivate_search, 5
 - deactivate_stale_searches, 6
 - handle_click, 8
 - list_searches, 10
 - send_search, 15
 - send_search_and_post, 16
- * **github**
 - dl_gh_repo, 6
 - get_branch_from_url, 7
 - get_subdir_from_url, 8
 - post_to_issue, 12
 - url_is_r_pkg, 21
- * **main**
 - collate_editor_check, 4
 - editor_check, 7
 - serve_api, 17
- * **pkgmatch**
 - cran_to_list, 5
 - pkgmatch_repo, 11
 - ros_to_list, 14
- * **ropensci**
 - check_issue_template, 3
 - is_user_authorized, 9
 - issue_is_stats, 10
 - push_to_gh_pages, 13
 - readme_has_peer_review_badge, 14
 - srr_counts, 17
 - srr_counts_from_report, 18
 - srr_counts_summary, 19
 - stats_badge, 19
- * **utils**
 - check_cache, 3
 - pkgrep_install_deps, 11
 - stdout_stderr_cache, 20
 - symbol_crs, 20
 - symbol_tck, 21

check_cache, 3
check_cache(), 12, 20, 21

check_issue_template, 3
check_issue_template(), 9, 10, 13, 14, 18–20
collate_editor_check, 4
collate_editor_check(), 7, 17
cran_to_list, 5
cran_to_list(), 11, 14

deactivate_search, 5
deactivate_search(), 6, 9, 10, 15, 16
deactivate_stale_searches, 6
deactivate_stale_searches(), 5, 9, 10, 15, 16

dl_gh_repo, 6
dl_gh_repo(), 8, 12, 21

editor_check, 7, 13, 20
editor_check(), 4, 17

get_branch_from_url, 7
get_branch_from_url(), 7, 8, 12, 21
get_subdir_from_url, 8
get_subdir_from_url(), 7, 8, 12, 21

handle_click, 8
handle_click(), 5, 6, 10, 15, 16

is_user_authorized, 9
is_user_authorized(), 4, 10, 13, 14, 18–20
issue_is_stats, 10
issue_is_stats(), 4, 9, 13, 14, 18–20

list_searches, 10
list_searches(), 5, 6, 9, 15, 16

pkgmatch_repo, 11
pkgmatch_repo(), 5, 14
pkgrep_install_deps, 11
pkgrep_install_deps(), 3, 20, 21
post_to_issue, 12
post_to_issue(), 7, 8, 21

push_to_gh_pages, 13
push_to_gh_pages(), 4, 9, 10, 14, 18–20

readme_badge, 13
readme_has_peer_review_badge, 14
readme_has_peer_review_badge(), 4, 9, 10, 13, 18–20

ros_to_list, 14
ros_to_list(), 5, 11

send_search, 15
send_search(), 5, 6, 9, 10, 16
send_search_and_post, 16
send_search_and_post(), 5, 6, 9, 10, 15

serve_api, 17
serve_api(), 4, 7

srr_counts, 17
srr_counts(), 4, 9, 10, 13, 14, 18–20
srr_counts_from_report, 18
srr_counts_from_report(), 4, 9, 10, 13, 14, 18–20

srr_counts_summary, 19
srr_counts_summary(), 4, 9, 10, 13, 14, 18, 20

stats_badge, 19
stats_badge(), 4, 9, 10, 13, 14, 18, 19

stdout_stderr_cache, 20
stdout_stderr_cache(), 3, 12, 20, 21

symbol_crs, 20
symbol_crs(), 3, 12, 20, 21

symbol_tck, 21
symbol_tck(), 3, 12, 20

url_is_r_pkg, 21
url_is_r_pkg(), 7, 8, 12