

Package: jinx (via r-universe)

May 25, 2026

Title RLadies+ GitHub Organisation Management

Version 0.1.1

Description GitHub App for managing the RLadies+ organisation. Handles onboarding and offboarding of global team members, chapter onboarding, PR review automation, org-wide reporting, Airtable sync, website automation, chapter monitoring, and interactive issue commands.

Depends R (>= 4.4.0)

License MIT + file LICENSE

URL <https://github.com/rladies/jinx>, <https://rladies.github.io/jinx/>

BugReports <https://github.com/rladies/jinx/issues>

Imports bskyr, cli, gh (>= 1.4.0), glue, httr2, jsonlite, jsonvalidate, lubridate, magick, openssl, rmarkdown, rtoot, rvest, stringr, xml2, yaml

Suggests httpptest2, knitr, pkgdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/Needs/website rladies/cloak, rprojroot

Config/pak/sysreqs cmake git make libmagick++-dev gsfonts libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev libx11-dev

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

Date/Publication 2026-05-25 20:57:28 UTC

RemoteUrl <https://github.com/rladies/jinx>

RemoteRef main

RemoteSha cbc3b80feddaf4879fb86b5614436270e263c800

Contents

analytics_collect_chapter_activity	4
analytics_collect_contributor_growth	4
analytics_compute_trends	5
analytics_format_markdown	5
analytics_generate_dashboard	6
analytics_publish_dashboard	6
announce_create_message	7
announce_post	7
announce_post_bluesky	8
announce_post_mastodon	8
announce_post_reply	9
announce_send_newsletter	9
blog_check_links	10
blog_create_entry	10
cfp_check_deadlines	11
cfp_create_issue	11
cfp_list_open	12
cfp_recommend_speaker	13
chapter_check_health	13
chapter_create	14
chapter_create_pr	15
chapter_create_setup	16
chapter_create_update	16
chapter_monitor_status	17
chapter_report_health	18
chunk_markdown	18
cloudflare_account_id	19
cloudflare_embed	19
cloudflare_vectorize_upsert	20
cmd_execute	20
cmd_parse	21
conference_generate_report	21
conference_list_speakers	22
contributor_format	22
contributor_list	23
contributor_list_org	23
contributor_update	24
directory_crop_image	24
directory_optimize_image	25
directory_sync_airtable	26
directory_validate_filename	26
directory_verify_handles	27
event_create_summary	27
event_list_chapter	28
event_publish_summary	28
event_sync_chapters	29

gather_rag_source	29
gh_greet_contributor	30
gh_post_checklist	30
gh_thank_contributor	31
gh_welcome_contributor	32
gha_generate_dashboard	33
gha_publish_dashboard	33
gt_check_invitations	34
gt_create_offboarding	34
gt_finalize_offboarding	35
gt_finalize_onboarding	35
gt_invite	36
gt_remind_stale	36
gt_sync_airtable	37
i18n_check_coverage	37
i18n_get_chapter_language	38
i18n_list_languages	38
i18n_translate_template	39
i18n_validate_translations	39
li_get_version	40
li_media_upload	40
li_oauth	41
li_post_write	41
li_req	42
li_urn_me	42
load_labels_config	43
load_rag_sources	43
load_review_rules	43
load_teams_config	44
prepare_inactivity_emails	44
rag_chunk_id	45
rag_index_build	45
random_emoji	46
report_format_markdown	46
report_generate	47
report_publish	47
review_check_pr_naming	48
review_run	48
short_url	49
slack_invite_batch	49
slack_invite_request	50
slack_post_message	51
slack_subscribe_rss	51
tags2hash	52
validate_blog_entry	52
validate_directory_entries	53
validate_directory_pr	53
website_collect_analytics	54

website_format_analytics	54
website_generate_report	55
website_merge_pending	55
website_publish_report	56

Index	57
--------------	-----------

analytics_collect_chapter_activity
Collect chapter activity data

Description

Queries the GitHub API for repository activity metrics across all non-meetup repos in the organization.

Usage

```
analytics_collect_chapter_activity(
  org = "rladies",
  months = 12,
  exclude_pattern = "^meetup-"
)
```

Arguments

org	GitHub organization.
months	Number of months of history.
exclude_pattern	Regex pattern to exclude repos.

Value

Data frame with columns: chapter, month, commits, prs, issues.

analytics_collect_contributor_growth
Collect contributor growth data

Description

Collect contributor growth data

Usage

```
analytics_collect_contributor_growth(org = "rladies", months = 12)
```

Arguments

org GitHub organization.
months Number of months of history.

Value

Data frame with columns: month, new_contributors, total_contributors, active_repos.

analytics_compute_trends
Compute activity trends

Description

Calculates month-over-month changes and rolling averages from chapter activity data.

Usage

```
analytics_compute_trends(activity_data)
```

Arguments

activity_data Data frame from [analytics_collect_chapter_activity\(\)](#).

Value

Data frame with trend columns added.

analytics_format_markdown
Format analytics as markdown

Description

Format analytics as markdown

Usage

```
analytics_format_markdown(trends, contributor_growth)
```

Arguments

trends Data frame from [analytics_compute_trends\(\)](#).
contributor_growth Data frame from [analytics_collect_contributor_growth\(\)](#).

Value

Formatted markdown string.

```
analytics_generate_dashboard
    Generate analytics dashboard
```

Description

Orchestrates data collection, trend computation, and formatting.

Usage

```
analytics_generate_dashboard(org = "rladies", months = 12, output_path = NULL)
```

Arguments

org	GitHub organization.
months	Number of months of history.
output_path	Optional path to write JSON data.

Value

Named list with trends, growth, and markdown (invisibly).

```
analytics_publish_dashboard
    Publish analytics dashboard as a GitHub issue
```

Description

Publish analytics dashboard as a GitHub issue

Usage

```
analytics_publish_dashboard(
  dashboard_data,
  org = "rladies",
  target_repo = "global-team",
  slack_channel = NULL
)
```

Arguments

dashboard_data Data from `analytics_generate_dashboard()`.
 org GitHub organization.
 target_repo Repository to publish to.
 slack_channel Optional Slack channel to post a summary to.

Value

Issue URL (invisibly).

announce_create_message

Create a formatted announcement message

Description

Create a formatted announcement message

Usage

```
announce_create_message(frontmatter, uri)
```

Arguments

frontmatter Named list from YAML front matter (needs title, description, tags).
 uri URL of the post.

Value

Formatted message string.

announce_post

Announce a blog post across multiple platforms

Description

Reads YAML front matter from a blog post, creates an announcement message, and posts to selected platforms.

Usage

```

announce_post(
  post,
  platforms = c("bluesky", "linkedin", "mastodon"),
  newsletter = TRUE
)

```

Arguments

post	File path to the blog post (markdown with YAML front matter).
platforms	Character vector of platforms to post to. Defaults to all: "bluesky", "linkedin", "mastodon".
newsletter	Logical, whether to also send a newsletter. Defaults to TRUE.

Value

Invisibly returns a named list with the posting result for each platform.

announce_post_bluesky *Post an announcement to Bluesky*

Description

Post an announcement to Bluesky

Usage

```
announce_post_bluesky(text, image = NULL, image_alt = NULL)
```

Arguments

text	Post text.
image	Optional file path to an image.
image_alt	Alt text for the image.

Value

Post response (invisibly).

announce_post_mastodon
Post an announcement to Mastodon

Description

Post an announcement to Mastodon

Usage

```
announce_post_mastodon(text, image = NULL, image_alt = NULL)
```

Arguments

text	Post text.
image	Optional file path to an image.
image_alt	Alt text for the image.

Value

Post response (invisibly).

announce_post_reply *Post a reply comment on an issue or PR*

Description

Posts as the authenticated user (jinx[bot] when using a GitHub App token).

Usage

```
announce_post_reply(owner, repo, issue_number, body)
```

Arguments

owner	Repository owner.
repo	Repository name.
issue_number	Issue or PR number.
body	Comment body (markdown).

Value

The API response (invisibly).

announce_send_newsletter
 Send a newsletter via ConvertKit

Description

Send a newsletter via ConvertKit

Usage

```
announce_send_newsletter(frontmatter, url)
```

Arguments

frontmatter Named list with seo, image, title fields.
 url URL of the post.

Value

API response (invisibly).

blog_check_links *Check blog URLs and RSS feeds for broken links*

Description

Check blog URLs and RSS feeds for broken links

Usage

blog_check_links(blogs_path)

Arguments

blogs_path Path to directory containing blog JSON files.

Value

A data frame with columns file, url, rss_feed, url_status, rss_status.

blog_create_entry *Auto-generate a blog entry JSON from a URL*

Description

Fetches OpenGraph metadata from the URL and creates a JSON entry compatible with awesome-rladies-blogs format.

Usage

blog_create_entry(url, language = "en", author_name, output_dir = ".")

Arguments

url Blog URL.
 language Language code. Defaults to "en".
 author_name Author name. Required.
 output_dir Directory to write the JSON file. Defaults to ".".

Value

File path of the created JSON (invisibly).

cfp_check_deadlines *Check CFP deadlines and post reminders*

Description

Check CFP deadlines and post reminders

Usage

```
cfp_check_deadlines(org = "rladies", repo = "global-team", warn_days = 7)
```

Arguments

org	GitHub organization.
repo	Repository where CFP issues are tracked.
warn_days	Number of days before deadline to warn.

Value

Data frame of approaching CFPs (invisibly).

cfp_create_issue *Create a CFP tracking issue*

Description

Create a CFP tracking issue

Usage

```
cfp_create_issue(  
  conference,  
  deadline,  
  url,  
  topics = character(0),  
  org = "rladies",  
  repo = "global-team"  
)
```

Arguments

conference	Conference name.
deadline	Submission deadline (YYYY-MM-DD).
url	CFP URL.
topics	Character vector of topic tags.
org	GitHub organization.
repo	Repository to create the issue in.

Value

Issue URL (invisibly).

cfp_list_open	<i>List open CFPs tracked as GitHub issues</i>
---------------	--

Description

List open CFPs tracked as GitHub issues

Usage

```
cfp_list_open(org = "rladies", repo = "global-team")
```

Arguments

org	GitHub organization.
repo	Repository where CFP issues are tracked.

Value

Data frame with columns: conference, deadline, url, number, status.

cfp_recommend_speaker *Recommend a speaker for a conference*

Description

Adds a speaker recommendation as a comment on the CFP tracking issue.

Usage

```
cfp_recommend_speaker(  
    conference,  
    speaker_name,  
    speaker_github = NULL,  
    expertise = character(0),  
    org = "rladies",  
    repo = "global-team"  
)
```

Arguments

conference	Conference name (matched against open CFP issues).
speaker_name	Speaker name or GitHub username.
speaker_github	Optional GitHub username.
expertise	Character vector of expertise areas.
org	GitHub organization.
repo	Repository where CFP issues are tracked.

Value

Comment URL (invisibly).

chapter_check_health *Check chapter health across the organization*

Description

Identifies chapters that have been inactive (no events) for a specified number of months.

Usage

```
chapter_check_health(  
    months = 12,  
    org = "rladies",  
    data_repo = "meetup_archive"  
)
```

Arguments

months	Number of months without activity to consider inactive. Defaults to 12.
org	GitHub organization. Defaults to "rladies".
data_repo	Repository containing chapter/event data. Defaults to "meetup_archive".

Value

A data frame with columns chapter, last_event, status, and months_inactive.

chapter_create	<i>Create a new chapter JSON file</i>
----------------	---------------------------------------

Description

Generates a chapter JSON entry for the website.

Usage

```
chapter_create(
  city,
  country,
  organizers,
  social_media = list(),
  status = "prospective",
  output_dir = "."
)
```

Arguments

city	City name.
country	Country name.
organizers	Character vector of organizer names.
social_media	Named list of social media links (e.g. list(meetup = "...", email = "city@rladies.org")).
status	Chapter status. Defaults to "prospective".
output_dir	Directory to write the JSON file.

Value

File path of the created JSON (invisibly).

chapter_create_pr *Create a chapter JSON PR on the website repo*

Description

Generates the chapter JSON file and creates a PR to add it to the website.

Usage

```
chapter_create_pr(  
  city,  
  country,  
  region = NULL,  
  meetup_urlname,  
  email,  
  organizers,  
  status = "prospective",  
  social_media = list(),  
  org = "rladies",  
  website_repo = "rladies.github.io"  
)
```

Arguments

city	Chapter city.
country	Chapter country.
region	State/region/province (optional).
meetup_urlname	Meetup group URL name.
email	Chapter email address.
organizers	Character vector of organizer names.
status	Chapter status. Defaults to "prospective".
social_media	Named list of social media handles (optional).
org	GitHub organization. Defaults to "rladies".
website_repo	Website repository name.

Value

PR URL (invisibly).

chapter_create_setup *Create a new chapter setup issue*

Description

Opens a tracking issue in the new-chapters-onboarding repo with the full checklist for setting up a new RLadies+ chapter.

Usage

```
chapter_create_setup(  
  city,  
  country,  
  organizers,  
  org = "rladies",  
  onboarding_repo = "new-chapters-onboarding"  
)
```

Arguments

city	Chapter city name.
country	Chapter country.
organizers	Character vector of organizer names.
org	GitHub organization. Defaults to "rladies".
onboarding_repo	Repository for chapter onboarding issues.

Value

Issue URL (invisibly).

chapter_create_update *Create a chapter update issue*

Description

Opens a tracking issue for updating an existing chapter's infrastructure.

Usage

```
chapter_create_update(  
  city,  
  country,  
  org = "rladies",  
  onboarding_repo = "new-chapters-onboarding"  
)
```

Arguments

city	Chapter city name.
country	Chapter country.
org	GitHub organization. Defaults to "rladies".
onboarding_repo	Repository for chapter onboarding issues.

Value

Issue URL (invisibly).

chapter_monitor_status
Monitor chapter activity status

Description

Determines chapter status based on meetup event data. Chapters are classified as active, inactive, or unbegun based on their last event and founding date.

Usage

```
chapter_monitor_status(data_path = NULL, inactive_months = 6, org = "rladies")
```

Arguments

data_path	Path to a CSV export from Meetup Pro Dashboard, or NULL to fetch from meetup_archive.
inactive_months	Months without events to consider inactive. Defaults to 6.
org	GitHub organization.

Value

Data frame with chapter status information.

chapter_report_health *Generate a chapter health report*

Description

Analyzes chapter activity data and publishes a summary report.

Usage

```
chapter_report_health(months = 6, org = "rladies", target_repo = "global-team")
```

Arguments

months	Inactivity threshold in months. Defaults to 6.
org	GitHub organization. Defaults to "rladies".
target_repo	Repository to publish report to. Defaults to "global-team".

Value

Issue URL (invisibly).

chunk_markdown *Chunk markdown into retrieval-sized pieces*

Description

Splits a markdown document by H1/H2 sections, then packs paragraphs into chunks of roughly target_chars characters. Returns a list of chunk records with heading, title, and lastmod metadata.

Usage

```
chunk_markdown(markdown, meta, target_chars = 1800, min_chars = 200)
```

Arguments

markdown	Markdown text (may include YAML frontmatter).
meta	Named list with repo, path, url, fallback_title, optional date, lastmod (both unix seconds).
target_chars	Approximate target chunk size.
min_chars	Minimum chunk size; smaller chunks are dropped.

Value

List of chunk records (one list per chunk).

cloudflare_account_id *Discover the Cloudflare account ID for a token*

Description

Fails if the token has access to zero or more than one account.

Usage

```
cloudflare_account_id(api_token)
```

Arguments

api_token Cloudflare API token.

Value

Account ID string.

cloudflare_embed *Embed texts with a Cloudflare Workers AI model*

Description

Calls the Cloudflare REST endpoint `accounts/{id}/ai/run/{model}` and returns the embedding vectors.

Usage

```
cloudflare_embed(  
  texts,  
  account_id,  
  api_token,  
  model = "@cf/baai/bge-base-en-v1.5"  
)
```

Arguments

texts Character vector of texts.
account_id Cloudflare account ID.
api_token Cloudflare API token.
model Workers AI embedding model.

Value

List of numeric vectors, one per input text.

cloudflare_vectorize_upsert
Upsert vectors into a Cloudflare Vectorize index

Description

Upsert vectors into a Cloudflare Vectorize index

Usage

```
cloudflare_vectorize_upsert(vectors, account_id, api_token, index_name)
```

Arguments

vectors	List of vector records, each with id, values, metadata.
account_id	Cloudflare account ID.
api_token	Cloudflare API token.
index_name	Vectorize index name.

Value

Parsed JSON response body.

cmd_execute *Execute a parsed jinx command*

Description

Returns the response message as a character string. The caller is responsible for routing the message to the right destination (GitHub issue comment, Slack, R console, etc.).

Usage

```
cmd_execute(command)
```

Arguments

command	Parsed command list from cmd_parse() .
---------	--

Value

Character string with the response message.

cmd_parse	<i>Parse a jinx command from an issue comment</i>
-----------	---

Description

Parses commands like:

- /jinx invite @username to website
- /jinx offboard @username from blog
- /jinx report weekly
- /jinx help

Usage

cmd_parse(body)

Arguments

body Character string, the comment body.

Value

A named list with action and action-specific fields, or NULL if the comment is not a jinx command.

conference_generate_report	<i>Generate conference coordination report</i>
----------------------------	--

Description

Summarizes active CFPs, speaker recommendations, and upcoming deadlines.

Usage

conference_generate_report(org = "rladies", repo = "global-team")

Arguments

org GitHub organization.
repo Repository where CFP issues are tracked.

Value

Formatted markdown string.

conference_list_speakers
List speaker recommendations for a conference

Description

List speaker recommendations for a conference

Usage

```
conference_list_speakers(conference, org = "rladies", repo = "global-team")
```

Arguments

conference	Conference name.
org	GitHub organization.
repo	Repository where CFP issues are tracked.

Value

Data frame with columns: speaker, expertise, recommended_by.

contributor_format *Generate a contributors markdown section*

Description

Creates a markdown table or image grid of contributors for inclusion in README or other documents.

Usage

```
contributor_format(contributors, format = c("table", "grid"), cols = 7)
```

Arguments

contributors	Data frame from contributor_list() .
format	Output format: "table" or "grid".
cols	Number of columns for grid format. Defaults to 7.

Value

Character string with markdown content.

contributor_list *List contributors for a repository*

Description

Collects all contributors from merged PRs and commits, returning a deduplicated list with contribution counts.

Usage

```
contributor_list(owner, repo, include_bots = FALSE)
```

Arguments

owner	Repository owner.
repo	Repository name.
include_bots	Whether to include bot accounts. Defaults to FALSE.

Value

Data frame with login, contributions, avatar_url, and profile_url columns, sorted by contribution count.

contributor_list_org *Collect contributors across multiple repos*

Description

Aggregates contributor data across all non-meetup repos in the org.

Usage

```
contributor_list_org(org = "rladies", exclude_pattern = "^meetup-")
```

Arguments

org	GitHub organization.
exclude_pattern	Regex to exclude repos.

Value

Data frame with login, repos, contributions, avatar_url, and profile_url. contributions is the sum across all repos; repos is the count of distinct repos contributed to.

contributor_update *Generate and update a contributors list for a repo*

Description

Fetches contributors and creates/updates a PR with the contributor list in a specified file.

Usage

```
contributor_update(  
  owner,  
  repo,  
  file_path = ".github/contributors.md",  
  format = "grid"  
)
```

Arguments

owner	Repository owner.
repo	Repository name.
file_path	Path in the repo to update with the contributor list. Defaults to ".github/contributors.md".
format	Format for the contributor list.

Value

PR URL if changes were made, NULL otherwise (invisibly).

directory_crop_image *Crop and resize a directory profile image*

Description

Uses magick to crop to a square and resize for web display.

Usage

```
directory_crop_image(  
  path,  
  width = 400,  
  height = 400,  
  gravity = "Center",  
  output = path  
)
```

Arguments

path	Path to the image file.
width	Target width in pixels. Defaults to 400.
height	Target height in pixels. Defaults to 400.
gravity	Crop gravity. Defaults to "Center".
output	Output path. Defaults to overwriting the input.

Value

Output path (invisibly).

directory_optimize_image

Optimize an image for web display

Description

Resizes to max width and compresses.

Usage

```
directory_optimize_image(path, max_width = 800, quality = 85, output = path)
```

Arguments

path	Path to the image file.
max_width	Maximum width in pixels. Defaults to 800.
quality	JPEG quality (1-100). Defaults to 85.
output	Output path. Defaults to overwriting the input.

Value

Output path (invisibly).

directory_sync_airtable

Sync directory entries from Airtable

Description

Fetches directory entries from Airtable and updates the directory repo. Creates a PR with new/updated entries for review.

Usage

```
directory_sync_airtable(  
  base_id = "appM6GuE0Jl1UI9qx",  
  api_key = Sys.getenv("AIRTABLE_API_KEY"),  
  org = "rladies",  
  directory_repo = "directory"  
)
```

Arguments

base_id	Airtable base ID.
api_key	Airtable API key. Defaults to AIRTABLE_API_KEY env var.
org	GitHub organization. Defaults to "rladies".
directory_repo	Directory repository name.

Value

PR URL if changes found, NULL otherwise (invisibly).

directory_validate_filename

Validate a directory entry filename

Description

Checks that filenames follow conventions: lowercase, no hashes, ASCII-only, .json extension.

Usage

```
directory_validate_filename(filename)
```

Arguments

filename	Filename (not full path) to validate.
----------	---------------------------------------

Value

A named list with `valid` (logical) and `issues` (character vector of problems found).

```
directory_verify_handles
```

Verify that social media handles exist

Description

Performs HTTP HEAD requests to check if profiles resolve.

Usage

```
directory_verify_handles(entry)
```

Arguments

`entry` Named list representing a directory entry, with a `social_media` sub-list containing handles/URLs.

Value

A data frame with columns `platform`, `handle`, `status`, and `valid`.

```
event_create_summary    Create a formatted event summary
```

Description

Create a formatted event summary

Usage

```
event_create_summary(events, period = c("weekly", "monthly"))
```

Arguments

`events` Data frame from [event_list_chapter\(\)](#) or [event_sync_chapters\(\)](#).
`period` Summary period label.

Value

Formatted markdown string.

event_list_chapter *List events for a chapter*

Description

Queries Meetup Pro for recent events.

Usage

```
event_list_chapter(chapter, months = 3)
```

Arguments

chapter Chapter identifier (e.g. "rladies-berlin").
months Number of months of history.

Value

Data frame with columns: title, date, url, rsvp_count, source, chapter.

event_publish_summary *Publish event summary as a GitHub issue*

Description

Publish event summary as a GitHub issue

Usage

```
event_publish_summary(summary, org = "rladies", target_repo = "global-team")
```

Arguments

summary Formatted summary from [event_create_summary\(\)](#).
org GitHub organization.
target_repo Repository to publish to.

Value

Issue URL (invisibly).

event_sync_chapters *Sync events across all chapters*

Description

Fetches events from all configured sources and writes a summary.

Usage

```
event_sync_chapters(
  org = "rladies",
  target_repo = "event-archive",
  months = 3,
  dry_run = TRUE
)
```

Arguments

org	GitHub organization.
target_repo	Repository for the event archive.
months	Number of months of history.
dry_run	If TRUE, print what would be synced without acting.

Value

Data frame of all events (invisibly).

gather_rag_source *Dispatch a source spec to the appropriate gather function*

Description

Looks up `gather_<type>()` where `<type>` has dashes replaced by underscores. Sources extend the indexer by defining a new `gather_<type>(src)` function.

Usage

```
gather_rag_source(src)
```

Arguments

src	Source spec list (must have type).
-----	------------------------------------

Value

List of chunk records (with `chunk_idx` set per document).

gh_greet_contributor *Greet a new PR author*

Description

Thin wrapper around `gh_welcome_contributor()` that fixes `is_pr = TRUE` for use in PR-only workflows. Reusable across any repo.

Usage

```
gh_greet_contributor(  
  owner,  
  repo,  
  number,  
  author,  
  org = "rladies",  
  extra_message = NULL  
)
```

Arguments

owner	Repository owner.
repo	Repository name.
number	PR number.
author	GitHub login of the author.
org	Organization name.
extra_message	Optional extra markdown to append after the standard welcome message.

Value

Comment URL or NULL if author is a team member (invisibly).

gh_post_checklist *Post a content review checklist on a PR*

Description

Posts a review checklist comment tailored for content PRs (blog posts, news posts, or any markdown-driven content). Generic across content kinds — the only difference between blog and news today is the path filter the caller workflow applies.

Usage

```
gh_post_checklist(owner, repo, pr_number)
```

Arguments

owner	Repository owner.
repo	Repository name.
pr_number	PR number.

Value

Comment URL (invisibly).

gh_thank_contributor *Thank a contributor when their PR is merged*

Description

Posts a thank-you message tailored to whether this is their first merged PR.

Usage

```
gh_thank_contributor(owner, repo, pr_number, author, org = "rladies")
```

Arguments

owner	Repository owner.
repo	Repository name.
pr_number	PR number.
author	GitHub login of the PR author.
org	Organization name.

Value

Comment URL (invisibly).

`gh_welcome_contributor`*Welcome a contributor on a new PR or issue*

Description

Checks if the PR/issue author has previous contributions to the repo. If this is their first, posts a warm welcome message. If not, posts a shorter thank-you. No-op for bots and global team members.

Usage

```
gh_welcome_contributor(  
    owner,  
    repo,  
    number,  
    author,  
    is_pr = TRUE,  
    org = "rladies",  
    extra_message = NULL  
)
```

Arguments

<code>owner</code>	Repository owner.
<code>repo</code>	Repository name.
<code>number</code>	Issue or PR number.
<code>author</code>	GitHub login of the author.
<code>is_pr</code>	Whether this is a PR (TRUE) or issue (FALSE).
<code>org</code>	Organization name.
<code>extra_message</code>	Optional extra markdown to append after the standard welcome message (e.g. a project-specific reminder). Ignored when blank or NULL. The jinx signature is preserved as the final paragraph.

Value

Comment URL or NULL if author is a team member (invisibly).

`gha_generate_dashboard`*Generate GitHub Actions dashboard data*

Description

Scans all non-meetup repositories in the organization, collects workflow information, and generates a JSON report.

Usage

```
gha_generate_dashboard(  
  org = "rladies",  
  exclude_pattern = "^meetup-",  
  output_path = NULL  
)
```

Arguments

`org` GitHub organization. Defaults to "rladies".

`exclude_pattern` Regex pattern to exclude repos. Defaults to "^meetup-".

`output_path` Path to write the JSON report. If NULL, returns the data without writing.

Value

List of workflow data (invisibly).

`gha_publish_dashboard` *Publish GHA dashboard as a GitHub issue*

Description

Creates a summary issue with workflow status badges.

Usage

```
gha_publish_dashboard(  
  dashboard_data,  
  org = "rladies",  
  target_repo = "global-team"  
)
```

Arguments

dashboard_data Data from `gha_generate_dashboard()`.
 org GitHub organization.
 target_repo Repository to publish to.

Value

Issue URL (invisibly).

gt_check_invitations *Check pending global team invitations*

Description

Lists pending invitations and triggers finalization for accepted ones.

Usage

```
gt_check_invitations(org = "rladies")
```

Arguments

org GitHub organization. Defaults to "rladies".

Value

Character vector of usernames that have accepted their invitation (invisibly).

gt_create_offboarding *Create a global team offboarding issue*

Description

Create a global team offboarding issue

Usage

```
gt_create_offboarding(username, team, name = username, org = "rladies")
```

Arguments

username GitHub username.
 team Team slug.
 name Full name. Defaults to the username.
 org GitHub organization. Defaults to "rladies".

Value

The created issue URL (invisibly).

`gt_finalize_offboarding`

Finalize global team offboarding by removing user from teams

Description

Finalize global team offboarding by removing user from teams

Usage

```
gt_finalize_offboarding(username, team, org = "rladies")
```

Arguments

username	GitHub username.
team	Team slug.
org	GitHub organization. Defaults to "rladies".

Value

Invisibly returns NULL. Called for its side effect of removing the user from the specified teams.

`gt_finalize_onboarding`

Finalize global team onboarding for an accepted member

Description

Adds the user to their specific team and creates a tracking issue with the onboarding checklist.

Usage

```
gt_finalize_onboarding(username, team, name = username, org = "rladies")
```

Arguments

username	GitHub username.
team	Team slug (e.g. "website").
name	Full name.
org	GitHub organization. Defaults to "rladies".

Value

The created issue URL (invisibly).

gt_invite	<i>Invite a user to the RLadies+ global team</i>
-----------	--

Description

Sends an organization membership invitation and adds the user to the global team.

Usage

```
gt_invite(username, team, name = username, org = "rladies")
```

Arguments

username	GitHub username (without @).
team	Team slug (e.g. "website").
name	Full name (used in issue templates).
org	GitHub organization. Defaults to "rladies".

Value

Invisibly returns NULL. Called for its side effect of sending the invitation and adding the user to the specified team.

gt_remind_stale	<i>Send reminders on stale global team onboarding/offboarding issues</i>
-----------------	--

Description

Finds open issues with onboarding or offboarding labels that haven't been updated in days days, and posts a reminder comment.

Usage

```
gt_remind_stale(org = "rladies", days = 30, repo = "global-team")
```

Arguments

org	GitHub organization. Defaults to "rladies".
days	Number of days without activity before reminding. Defaults to 30.
repo	Repository to check. Defaults to "global-team".

Value

Invisibly returns NULL. Called for its side effect of posting reminder comments on stale issues.

gt_sync_airtable	<i>Sync global team data from Airtable</i>
------------------	--

Description

Fetches global team member data from Airtable and updates the website repo data files.

Usage

```
gt_sync_airtable(  
  base_id = "appZjaV7eM0Y9FsHZ",  
  api_key = Sys.getenv("AIRTABLE_API_KEY"),  
  org = "rladies",  
  website_repo = "rladies.github.io"  
)
```

Arguments

base_id	Airtable base ID for the global team.
api_key	Airtable API key.
org	GitHub organization.
website_repo	Website repository name.

Value

PR URL if changes found, NULL otherwise (invisibly).

i18n_check_coverage	<i>Check translation coverage across languages</i>
---------------------	--

Description

Check translation coverage across languages

Usage

```
i18n_check_coverage()
```

Value

Data frame with columns: language, total_templates, translated, coverage_pct.

i18n_get_chapter_language

Get a chapter's preferred language

Description

Looks up the chapter's language preference from its metadata. Falls back to English if not set.

Usage

```
i18n_get_chapter_language(chapter, org = "rladies")
```

Arguments

chapter	Chapter slug (e.g. "rladies-berlin").
org	GitHub organization.

Value

Language code string.

i18n_list_languages *List supported languages*

Description

List supported languages

Usage

```
i18n_list_languages()
```

Value

Data frame with columns: code, name, native_name, direction.

`i18n_translate_template`*Translate a template with language fallback*

Description

Looks up `inst/translations/{language}/{template_name}.md` first, falls back to `inst/templates/{template_name}` if the translation is missing.

Usage

```
i18n_translate_template(template_name, language = "en", variables = list())
```

Arguments

<code>template_name</code>	Template filename without path (e.g. "global-team-onboarding.md").
<code>language</code>	Language code (e.g. "es", "pt", "fr"). Defaults to "en".
<code>variables</code>	Named list of placeholder values for <code>render_template()</code> .

Value

Rendered template as a single character string.

`i18n_validate_translations`*Validate translations for placeholder consistency*

Description

Checks that all translated templates have the same `<KEY>` placeholders as the English baseline.

Usage

```
i18n_validate_translations(language = NULL)
```

Arguments

<code>language</code>	Language code to validate. If NULL, validates all.
-----------------------	--

Value

Data frame with columns: `template`, `language`, `status`, `missing_keys`, `extra_keys`.

li_get_version	<i>Get LinkedIn API version string</i>
----------------	--

Description

Get LinkedIn API version string

Usage

```
li_get_version()
```

Value

Version string in YYYYMM format.

li_media_upload	<i>Upload media to LinkedIn</i>
-----------------	---------------------------------

Description

Upload media to LinkedIn

Usage

```
li_media_upload(author, media)
```

Arguments

author	LinkedIn URN of the owner.
media	File path to the image.

Value

LinkedIn image ID string.

li_oauth	<i>Perform LinkedIn OAuth authentication</i>
----------	--

Description

Interactive OAuth2 flow for obtaining a LinkedIn token.

Usage

```
li_oauth()
```

Value

OAuth token object.

li_post_write	<i>Post to LinkedIn</i>
---------------	-------------------------

Description

Post to LinkedIn

Usage

```
li_post_write(author, text, image = NULL, image_alt = "")
```

Arguments

author	LinkedIn URN of the author.
text	Post text.
image	Optional file path to an image.
image_alt	Alt text for the image.

Value

LinkedIn post ID (invisibly).

li_req	<i>Create a base LinkedIn API request</i>
--------	---

Description

Create a base LinkedIn API request

Usage

```
li_req(endpoint_version = "rest", ...)
```

Arguments

endpoint_version	API version path. Defaults to "rest".
...	Passed to li_req_auth().

Value

Configured httr2 request.

li_urn_me	<i>Get the LinkedIn URN for the authenticated user</i>
-----------	--

Description

Get the LinkedIn URN for the authenticated user

Usage

```
li_urn_me()
```

Value

URN string like "urn:li:person:abc123".

load_labels_config *Load file-path to label mappings*

Description

Load file-path to label mappings

Usage

load_labels_config()

Value

Named list with label mappings.

load_rag_sources *Load the configured RAG source list*

Description

Load the configured RAG source list

Usage

load_rag_sources()

Value

List of source specs.

load_review_rules *Load PR review rules*

Description

Load PR review rules

Usage

load_review_rules()

Value

Named list with defaults, rules, and naming conventions.

load_teams_config	<i>Load teams configuration</i>
-------------------	---------------------------------

Description

Load teams configuration

Usage

```
load_teams_config()
```

Value

Named list with organization, global_team_id, teams, and default_assignees.

prepare_inactivity_emails	<i>Send inactivity warning emails</i>
---------------------------	---------------------------------------

Description

Identifies inactive chapters and prepares warning emails for organizers.

Usage

```
prepare_inactivity_emails(chapters, template_path = NULL, dry_run = TRUE)
```

Arguments

chapters	Chapter status data from chapter_monitor_status() .
template_path	Path to email template.
dry_run	If TRUE (default), only returns the email data without sending.

Value

Data frame of prepared emails (invisibly).

rag_chunk_id	<i>Stable vector ID for a chunk</i>
--------------	-------------------------------------

Description

Matches the JS indexer scheme: first 32 hex chars of sha256("{repo}|{path}|{chunk_idx}").

Usage

```
rag_chunk_id(repo, path, chunk_idx)
```

Arguments

repo	Source repo (e.g. "rladies/rladiesguide").
path	Path within the source (e.g. /getting-started/).
chunk_idx	Zero-based chunk index within the document.

Value

32-character hex string.

rag_index_build	<i>Build the Jinx RAG index</i>
-----------------	---------------------------------

Description

Gathers chunks from every configured source, embeds them with Cloudflare Workers AI, and upserts the vectors into a Vectorize index. The source list is loaded from `inst/config/rag-sources.yml` unless `sources` is supplied directly.

Usage

```
rag_index_build(
  sources = NULL,
  account_id = NULL,
  api_token = Sys.getenv("CLOUDFLARE_API_TOKEN"),
  index_name = NULL,
  batch_size = 50L,
  model = "@cf/baai/bge-base-en-v1.5"
)
```

Arguments

sources	Optional list of source specs (see <code>load_rag_sources()</code>).
account_id	Cloudflare account ID. Falls back to env <code>CLOUDFLARE_ACCOUNT_ID</code> , then <code>cloudflare_account_id()</code> .
api_token	Cloudflare API token. Defaults to env <code>CLOUDFLARE_API_TOKEN</code> .
index_name	Vectorize index. Defaults to env <code>VECTORIZE_INDEX</code> or "rladies-content".
batch_size	Number of chunks per embed/upsert call.
model	Workers AI embedding model.

Value

Invisibly: list with chunks (total) and upsert (API response).

random_emoji	<i>Select a random emoji</i>
--------------	------------------------------

Description

Select a random emoji

Usage

```
random_emoji()
```

Value

Single emoji character.

report_format_markdown	<i>Format a report as markdown</i>
------------------------	------------------------------------

Description

Format a report as markdown

Usage

```
report_format_markdown(report)
```

Arguments

report	Report data from <code>report_generate()</code> .
--------	---

Value

Character string with markdown-formatted report.

report_generate	<i>Generate an organization activity report</i>
-----------------	---

Description

Collects stats across org repos: commits, PRs, issues.

Usage

```
report_generate(
  type = c("weekly", "monthly"),
  org = "rladies",
  exclude_pattern = "^meetup-"
)
```

Arguments

type	Report type: "weekly" or "monthly".
org	GitHub organization. Defaults to "rladies".
exclude_pattern	Regex to exclude repos. Defaults to "^meetup-".

Value

A named list with report data (invisibly).

report_publish	<i>Publish a report as a GitHub issue</i>
----------------	---

Description

Publish a report as a GitHub issue

Usage

```
report_publish(report, target_repo = "global-team", org = "rladies")
```

Arguments

report	Report data from report_generate() .
target_repo	Repository to publish to. Defaults to "global-team".
org	Organization. Defaults to "rladies".

Value

Issue URL (invisibly).

```
review_check_pr_naming
```

Check file naming conventions in a PR

Description

Validates that changed files follow repository naming conventions.

Usage

```
review_check_pr_naming(owner, repo, pr_number)
```

Arguments

owner	Repository owner.
repo	Repository name.
pr_number	PR number.

Value

Data frame with filename and valid columns.

```
review_run
```

Run all PR review automation

Description

Assigns reviewers, labels the PR, and posts a checklist comment.

Usage

```
review_run(owner, repo, pr_number)
```

Arguments

owner	Repository owner.
repo	Repository name.
pr_number	Pull request number.

Value

Invisibly returns NULL. Called for its side effects of assigning reviewers, labelling the PR, and posting a checklist comment.

short_url	<i>Shorten a URL using the Short.io API</i>
-----------	---

Description

Shorten a URL using the Short.io API

Usage

```
short_url(uri)
```

Arguments

uri	URL to shorten.
-----	-----------------

Value

Shortened URL string.

slack_invite_batch	<i>Send pending Slack invitations</i>
--------------------	---------------------------------------

Description

Fetches pending invitees from Airtable and prepares invite emails with a Slack invitation link.

Usage

```
slack_invite_batch(
  invite_link,
  base_id = slack_invitees_base_id(),
  api_key = Sys.getenv("AIRTABLE_API_KEY"),
  sender_name = "RLadies+ Global",
  sender_email = "info@rladies.org",
  dry_run = TRUE
)
```

Arguments

invite_link	Active Slack invite URL (valid for 30 days).
base_id	Airtable base ID for Slack invitees.
api_key	Airtable API key.
sender_name	Name of the sender.
sender_email	Sender email address.
dry_run	If TRUE (default), only returns prepared emails.

Value

Data frame of prepared emails (invisibly).

slack_invite_request *Request a Slack invitation for someone not yet on the workspace*

Description

The bot cannot invite users directly (Slack admin scopes are not available to community apps). Instead, this function posts a message to an organisers' Slack channel with the email address and instructions for a workspace admin to action the invite manually, then flips the matching Airtable record's invited flag for audit.

Usage

```
slack_invite_request(
  email,
  channel = Sys.getenv("SLACK_INVITE_REQUEST_CHANNEL"),
  base_id = slack_invitees_base_id(),
  api_key = Sys.getenv("AIRTABLE_API_KEY"),
  token = Sys.getenv("SLACK_TOKEN")
)
```

Arguments

email	Email address to request an invite for.
channel	Slack channel where the request is posted. Defaults to SLACK_INVITE_REQUEST_CHANNEL env var, falling back to "global-team".
base_id	Airtable base ID for Slack invitees.
api_key	Airtable API key. If unset, the Airtable audit step is skipped.
token	Slack bot token (chat:write scope is enough).

Value

A single status string suitable for posting back as a PR comment.

slack_post_message *Post a message to a Slack channel*

Description

Sends a markdown-formatted message to the specified Slack channel using the Slack Web API.

Usage

```
slack_post_message(text, channel, token = Sys.getenv("SLACK_TOKEN"))
```

Arguments

text	Message text (supports Slack mrkdwn formatting).
channel	Slack channel name (without #).
token	Slack API token. Defaults to Sys.getenv("SLACK_TOKEN").

Value

API response (invisibly).

slack_subscribe_rss *Subscribe an RSS feed to a Slack channel*

Description

Posts a /feed subscribe command to the specified Slack channel using the Slack API.

Usage

```
slack_subscribe_rss(  
  rss_url,  
  channel = "rladiesblogs",  
  token = Sys.getenv("SLACK_TOKEN")  
)
```

Arguments

rss_url	RSS feed URL to subscribe.
channel	Slack channel name (without #). Defaults to "rladiesblogs".
token	Slack API token.

Value

API response (invisibly).

tags2hash	<i>Convert tags to hashtag string</i>
-----------	---------------------------------------

Description

Convert tags to hashtag string

Usage

```
tags2hash(tags)
```

Arguments

tags	Character vector of tags.
------	---------------------------

Value

Single string with hashtags.

validate_blog_entry	<i>Validate blog entry JSON files against schema</i>
---------------------	--

Description

Validate blog entry JSON files against schema

Usage

```
validate_blog_entry(  
  path,  
  schema = system.file("schemas", "blog-entry.json", package = "jinx")  
)
```

Arguments

path	Path to a directory containing blog JSON files, or a single JSON file path.
schema	Path to JSON schema file. Uses the bundled schema by default.

Value

A data frame with columns file, valid, and errors.

`validate_directory_entries`*Validate directory entry JSON files against schema*

Description

Validate directory entry JSON files against schema

Usage

```
validate_directory_entries(  
    path,  
    schema = system.file("schemas", "directory-entry.json", package = "jinx")  
)
```

Arguments

<code>path</code>	Path to a directory containing JSON entry files, or a single JSON file path.
<code>schema</code>	Path to JSON schema file. Uses the bundled schema by default.

Value

A data frame with columns `file`, `valid`, and `errors`.

`validate_directory_pr` *Post a directory validation report as a PR comment*

Description

Runs all directory validations (schema, filenames, social handles) on changed files in a PR and posts a summary comment.

Usage

```
validate_directory_pr(owner, repo, pr_number)
```

Arguments

<code>owner</code>	Repository owner.
<code>repo</code>	Repository name.
<code>pr_number</code>	PR number.

Value

Invisibly returns the URL of the posted PR comment.

```
website_collect_analytics
```

Collect website analytics from Plausible

Description

Queries the Plausible Analytics API for visitor and pageview metrics.

Usage

```
website_collect_analytics(
  site_id = Sys.getenv("PLAUSIBLE_SITE_ID"),
  api_key = Sys.getenv("PLAUSIBLE_API_KEY"),
  base_url = Sys.getenv("PLAUSIBLE_URL", "https://plausible.io"),
  period = c("30d", "7d", "month", "6mo", "12mo")
)
```

Arguments

site_id	Plausible site ID (domain). Defaults to Sys.getenv("PLAUSIBLE_SITE_ID").
api_key	Plausible API key. Defaults to Sys.getenv("PLAUSIBLE_API_KEY").
base_url	Plausible instance URL. Defaults to Sys.getenv("PLAUSIBLE_URL", "https://plausible.io").
period	Time period: "30d", "7d", "month", "6mo", "12mo".

Value

Named list with aggregate, timeseries, top_pages, and top_sources.

```
website_format_analytics
```

Format website analytics as markdown

Description

Format website analytics as markdown

Usage

```
website_format_analytics(analytics)
```

Arguments

analytics	Data from website_collect_analytics() .
-----------	---

Value

Formatted markdown string.

 website_generate_report

Generate a website analytics report

Description

Collects Plausible analytics and formats as markdown.

Usage

```
website_generate_report(
  site_id = Sys.getenv("PLAUSIBLE_SITE_ID"),
  api_key = Sys.getenv("PLAUSIBLE_API_KEY"),
  base_url = Sys.getenv("PLAUSIBLE_URL", "https://plausible.io"),
  period = c("30d", "7d", "month", "6mo", "12mo"),
  output_path = NULL
)
```

Arguments

site_id	Plausible site ID (domain). Defaults to Sys.getenv("PLAUSIBLE_SITE_ID").
api_key	Plausible API key. Defaults to Sys.getenv("PLAUSIBLE_API_KEY").
base_url	Plausible instance URL. Defaults to Sys.getenv("PLAUSIBLE_URL", "https://plausible.io").
period	Time period: "30d", "7d", "month", "6mo", "12mo".
output_path	Optional path to write JSON data.

Value

Named list with analytics and markdown (invisibly).

 website_merge_pending *Auto-merge PRs with pending label when date matches*

Description

Checks open PRs with "pending" label in the website repo. If a blog post's date matches today, merges the PR.

Usage

```
website_merge_pending(org = "rladies", repo = "rladies.github.io")
```

Arguments

org	GitHub organization.
repo	Repository name.

Value

Character vector of merged PR URLs (invisibly).

website_publish_report

Publish website analytics as a GitHub issue

Description

Publish website analytics as a GitHub issue

Usage

```
website_publish_report(  
  report_data,  
  org = "rladies",  
  target_repo = "global-team",  
  slack_channel = NULL  
)
```

Arguments

report_data	Data from website_generate_report() .
org	GitHub organization.
target_repo	Repository to publish to.
slack_channel	Optional Slack channel to post a summary to.

Value

Issue URL (invisibly).

Index

analytics_collect_chapter_activity, 4
analytics_collect_chapter_activity(), 5
analytics_collect_contributor_growth, 4
analytics_collect_contributor_growth(), 5
analytics_compute_trends, 5
analytics_compute_trends(), 5
analytics_format_markdown, 5
analytics_generate_dashboard, 6
analytics_generate_dashboard(), 7
analytics_publish_dashboard, 6
announce_create_message, 7
announce_post, 7
announce_post_bluesky, 8
announce_post_mastodon, 8
announce_post_reply, 9
announce_send_newsletter, 9

blog_check_links, 10
blog_create_entry, 10

cfp_check_deadlines, 11
cfp_create_issue, 11
cfp_list_open, 12
cfp_recommend_speaker, 13
chapter_check_health, 13
chapter_create, 14
chapter_create_pr, 15
chapter_create_setup, 16
chapter_create_update, 16
chapter_monitor_status, 17
chapter_monitor_status(), 44
chapter_report_health, 18
chunk_markdown, 18
cloudflare_account_id, 19
cloudflare_embed, 19
cloudflare_vectorize_upsert, 20
cmd_execute, 20
cmd_parse, 21
cmd_parse(), 20
conference_generate_report, 21
conference_list_speakers, 22
contributor_format, 22
contributor_list, 23
contributor_list(), 22
contributor_list_org, 23
contributor_update, 24

directory_crop_image, 24
directory_optimize_image, 25
directory_sync_airtable, 26
directory_validate_filename, 26
directory_verify_handles, 27

event_create_summary, 27
event_create_summary(), 28
event_list_chapter, 28
event_list_chapter(), 27
event_publish_summary, 28
event_sync_chapters, 29
event_sync_chapters(), 27

gather_rag_source, 29
gh_greet_contributor, 30
gh_post_checklist, 30
gh_thank_contributor, 31
gh_welcome_contributor, 32
gh_welcome_contributor(), 30
gha_generate_dashboard, 33
gha_generate_dashboard(), 34
gha_publish_dashboard, 33
gt_check_invitations, 34
gt_create_offboarding, 34
gt_finalize_offboarding, 35
gt_finalize_onboarding, 35
gt_invite, 36
gt_remind_stale, 36
gt_sync_airtable, 37

- i18n_check_coverage, 37
- i18n_get_chapter_language, 38
- i18n_list_languages, 38
- i18n_translate_template, 39
- i18n_validate_translations, 39

- li_get_version, 40
- li_media_upload, 40
- li_oauth, 41
- li_post_write, 41
- li_req, 42
- li_urn_me, 42
- load_labels_config, 43
- load_rag_sources, 43
- load_review_rules, 43
- load_teams_config, 44

- prepare_inactivity_emails, 44

- rag_chunk_id, 45
- rag_index_build, 45
- random_emoji, 46
- report_format_markdown, 46
- report_generate, 47
- report_generate(), 46, 47
- report_publish, 47
- review_check_pr_naming, 48
- review_run, 48

- short_url, 49
- slack_invite_batch, 49
- slack_invite_request, 50
- slack_post_message, 51
- slack_subscribe_rss, 51

- tags2hash, 52

- validate_blog_entry, 52
- validate_directory_entries, 53
- validate_directory_pr, 53

- website_collect_analytics, 54
- website_collect_analytics(), 54
- website_format_analytics, 54
- website_generate_report, 55
- website_generate_report(), 56
- website_merge_pending, 55
- website_publish_report, 56