aiohttp_github_helpers Documentation

Fabien MARTY

Sep 11, 2020

Contents

aiohttp_github_helpers.**github_add_labels_on_issue**(*client_session*, *owner*, *repo*, *issue_number*, *labels_to_add*)

Add somes labels to a github issue.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **issue_number** (*int*) – number of the issue at github.
- **labels_to_add** (*list*) – list of strings to add as labels.

Returns True if it's ok, False else

Return type boolean

aiohttp_github_helpers.**github_check_github_event**(*request*, *handler*)

Check the GitHub event or return an HTTP/400.

This is an aiohttp middleware. If we can't get the X-GitHub-Event in request headers, we return an HTTP/400 error. Else, the value is stored in the request object dict in the key 'github_event'.

Parameters

- **request** – aiohttp.web.Request object corresponding to the incoming http request from the client.
- **handler** – aiohttp handler (see middlewares documentation).

Returns aiohttp Response (see middlewares documentation).

aiohttp_github_helpers.**github_check_signature_middleware_factory**(*signature_secret*)

Build and return an aiohttp middleware to check the GitHub signature.

We check the GitHub hook signature (see corresponding doc at github on hook secrets). If it doesn't match to the given signature_secret, we return a HTTP/400 error.

Parameters **signature_secret** (*bytes*) – the signature secret (as bytes).

Returns aiohttp middlerware (see aiohttp middlewares documentation).

aiohttp_github_helpers.**github_conditional_add_label_on_issue**(*client_session*, *owner*, *repo*, *issue_number*, *label_to_add*, *glob_not_to_match*)

Add label to a github issue if none on current labels match the given glob.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **issue_number** (*int*) – number of the issue at github.
- **lable_to_add** (*string*) – the label to add.
- **glob_not_to_match** (*string*) – glob string (as defined in fnmatch module) to test with every current labels on the issue.

Returns True if the label was added, False else.

Return type boolean

aiohttp_github_helpers.**github_create_status**(*client_session*, *owner*, *repo*, *sha*, *status_state*, *status_target_url*, *status_description*, *status_context*)

Create a status for the given sha.

see https://developer.github.com/v3/repos/statuses/#create-a-status

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **sha** (*string*) – the sha of the commit.
- **status_state** (*string*) – the state of the status (error, failure, pending or success).
- **status_target_url** – status target url (see github doc).
- **status_description** – status description (see github doc).
- **status_context** – status context (see github doc).

Returns True if the status was created, False else.

Return type boolean

aiohttp_github_helpers.**github_delete_label_on_issue**(*client_session*, *owner*, *repo*, *issue_number*, *label*)

Delete a label from a github issue.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **issue_number** (*int*) – number of the issue at github.
- **label** (*string*) – label string to delete

Returns True if ok, False else

Return type boolean

aiohttp_github_helpers.**github_delete_labels_on_issue_with_globs**(*client_session*, *owner*, *repo*, *issue_number*, *glob_include*, *glob_exclude=''*)

Delete some labels from a github issue with globs (see fnmatch module).

To be deleted a label must match to the glob_include pattern AND NOT match to the glob_exclude pattern.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **issue_number** (*int*) – number of the issue at github.

- **glob_include** (*string*) – glob string (as defined in fnmatch module) to select which labels to delete.
- **glob_exclude** (*string*) – glob string (as defined in fnmatch module) to select which labels NOT to delete.

Returns

tuple of lists. The first element is the list of remaining labels (or None if problems), Second element is the original list of labels (or None if problems).

Return type (list, list)

aiohttp_github_helpers.**github_get_labels_on_issue**(*client_session*, *owner*, *repo*, *issue_number*)

Get the list of labels of a github issue.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **issue_number** (*int*) – number of the issue at github.

Returns list of labels (as strings) or None (if problems).

Return type list

aiohttp_github_helpers.**github_get_latest_commit**(*client_session*, *owner*, *repo*, *branch*, *timezone='Europe/Paris'*)

Get the latest commit for a particular branch on a repo.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **branch** (*string*) – the branch name.
- **timezone** (*string*) – the current timezone.

Returns ("sha", age_in_seconds) or None if problem.

Return type latest commit as a tuple

aiohttp_github_helpers.**github_get_open_prs_by_sha**(*client_session*, *owner*, *repo*, *sha*, *state='open'*)

Get the list of pr where head is the given sha.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **sha** (*string*) – the SHA to search.
- **state** (*string*) – either open, closed, all to filter by pr state.

Returns pr numbers as list of int

Return type pr numbers (list)

aiohttp_github_helpers.**github_get_org_repos_by_topic**(*client_session*, *org*, *topics_to_include=None*, *topics_to_excludes=[]*)

Get the repo names of an organization by filtering by topics.

Parameters

- **client_session** – aiohttp ClientSession.
- **org** – organization name
- **topics_to_include** (*list*) – list of topics (AND) each repo must have (if None, no filtering).
- **topics_to_exlude** (*list*) – list of topics (OR) each repo must not have to be in result.

Returns list or repo names.

Return type repo names (list)

aiohttp_github_helpers.**github_get_pr_commit_messages_list**(*client_session*, *owner*, *repo*, *pr_number*)

Get the list of commit messages for a given pull-request.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **pr_number** (*int*) – the pull-request number.

Returns

list of commit messages (of the PR) as strings (or None if problems)

Return type list

aiohttp_github_helpers.**github_get_pr_reviews**(*client_session*, *owner*, *repo*, *pr_number*)

Get the reviews of a given pull-request.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **pr_number** (*int*) – the pull-request number.

Returns

list of reviews (each review is a dict with user_login, state, and sha keys)

Return type list

aiohttp_github_helpers.**github_get_repo_topics**(*client_session*, *owner*, *repo*)

Get the topics of a repository.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).

Returns list of repo topics.

Return type topics (list)

aiohttp_github_helpers.**github_get_status**(*client_session*, *owner*, *repo*, *ref*, *ignore_context_globs=[]*)

Get the combined status for a given ref.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **ref** (*string*) – the ref can be a SHA, a branch name, or a tag name.
- **ignore_context_globs** (*list*) – list of context to ignore (globs as defined by fnmatch module).

Returns combined state (failure, success...)

Return type combined state (string)

aiohttp_github_helpers.**github_post_comment**(*client_session*, *owner*, *repo*, *issue_number*, *comment_body*)

Post a comment to a github issue.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **issue_number** (*int*) – the issue number.
- **comment_body** (*string*) – the body of the comment.

Returns True if the comment was created, False else.

Return type boolean

aiohttp_github_helpers.**github_replace_labels_with**(*client_session*, *owner*, *repo*, *issue_number*, *glob_to_remove*, *new_label*, *always_add=False*)

Replace some labels from a github issue matching with a glob by a new one.

Note: if the new_label to add is already present, it is not removed then added another time.

Parameters

- **client_session** – aiohttp ClientSession.
- **owner** – owner of the repository at github.
- **repo** – repository name at github (without owner part).
- **issue_number** (*int*) – number of the issue at github.
- **glob_to_remove** (*string*) – glob string (as defined in fnmatch module) to select which labels to delete.
- **new_label** (*string*) – new label to add.
- **always_add** (*boolean*) – True if you want to add the new_label even if no label was removed.

Returns

True if new_label was added (or was already here), False else, None if problems.

Return type boolean

Python Module Index

a

Index

A

G