
PyGithub Documentation

Release 1.40

Vincent Jacques

Jul 15, 2018

Contents

| | | |
|----------|----------------------------|------------|
| 1 | Introduction | 1 |
| 2 | Reference | 5 |
| 3 | Change log | 113 |
| | Python Module Index | 129 |

PyGithub is a Python (2 and 3) library to use the [Github API v3](#). With it, you can manage your [Github](#) resources (repositories, user profiles, organizations, etc.) from Python scripts.

Should you have any question, any remark, or if you find a bug, or if there is something you can do with the API but not with PyGithub, please [open an issue](#).

1.1 (Very short) tutorial

First create a Github instance:

```
from github import Github

# using username and password
g = Github("user", "password")

# or using an access token
g = Github("access_token")
```

Then play with your Github objects:

```
for repo in g.get_user().get_repos():
    print(repo.name)
    repo.edit(has_wiki=False)
```

1.2 Download and install

This package is in the [Python Package Index](#), so `pip install PyGithub` should be enough. You can also clone it on [Github](#).

If you wish to use [GitHub Integrations](#), you'll want to be sure to install the 'integrations' option: `pip install PyGithub['integrations']`

1.3 Licensing

PyGithub is distributed under the GNU Lesser General Public Licence. See files COPYING and COPYING.LESSER, as requested by GNU.

1.4 What next?

You need to use a Github API and wonder which class implements it? [Reference of APIs](#).

You want all the details about PyGithub classes? [Reference of Classes](#).

1.5 Projects using PyGithub

(Open an issue if you want to be listed here, I'll be glad to add your project)

- [Github-iCalendar](#) returns all of your Github issues and pull requests as a list of tasks / VTOD items in iCalendar format.
- [DevAssistant](#)
- [Upverter](#) is a web-based schematic capture and PCB layout tool for people who design electronics. Designers can attach a Github project to an Upverter project.
- [Notifico](#) receives messages (such as commits and issues) from services and scripts and delivers them to IRC channels. It can import/sync from Github.
- [Tratihubis](#) converts Trac tickets to Github issues
- <https://github.com/fga-gpp-mds/2018.1-Cardinals> - website that shows metrics for any public repository (issues, commits, pull requests etc)
- <https://github.com/CMB/cligh>
- <https://github.com/natduca/quickopen> uses PyGithub to automatically create issues
- <https://gist.github.com/3433798>
- <https://github.com/zsiciarz/aquila-dsp.org>
- <https://github.com/robcowie/virtualenvwrapper.github>
- <https://github.com/kokosing/git-gifi> - Git and github enhancements to git.
- <https://github.com/csurfer/gitsuggest> - A tool to suggest github repositories based on the repositories you have shown interest in
- <https://github.com/gomesfernanda/some-github-metrics> - Python functions for relevant metrics on GitHub repositories
- <https://github.com/SOM-Research/Gitana> - a SQL-based Project Activity Inspector
- <https://github.com/YakDriver/satsukia> - Python package that helps manage GitHub releases and release assets
- [check-in](#) — Python CLI distribution that allows user to use GitHub Checks API as a bot.

1.6 They talk about PyGithub

- <http://stackoverflow.com/questions/10625190/most-suitable-python-library-for-github-api-v3>
- <http://stackoverflow.com/questions/12379637/django-social-auth-github-authentication>
- <http://www.freebsd.org/cgi/cvsweb.cgi/ports/devel/py-pygithub/>
- https://bugzilla.redhat.com/show_bug.cgi?id=910565

The primary class you will instantiate is `github.MainClass.Github`. From its `get_`, `create_` methods, you will obtain instances of all Github objects like `github.NamedUser.NamedUser` or `github.Repository.Repository`.

All classes inherit from `github.GithubObject.GithubObject`.

2.1 Main class: Github

```
class github.MainClass.Github (login_or_token=None, password=None,
                                base_url='https://api.github.com', timeout=15, client_id=None,
                                client_secret=None, user_agent='PyGithub/Python',
                                per_page=30, api_preview=False, verify=True)
```

This is the main class you instantiate to access the Github API v3. Optional parameters allow different authentication methods.

Parameters

- **login_or_token** – string
- **password** – string
- **base_url** – string
- **timeout** – integer
- **client_id** – string
- **client_secret** – string
- **user_agent** – string
- **per_page** – int
- **verify** – boolean or string

FIX_REPO_GET_GIT_REF

Type bool

per_page

Type int

rate_limiting

First value is requests remaining, second value is request limit. :type: (int, int)

rate_limiting_resettime

Unix timestamp indicating when rate limiting will reset. :type: int

get_rate_limit()

Don't forget you can access the rate limit returned in headers of last Github API v3 response, by `github.MainClass.Github.rate_limiting` and `github.MainClass.Github.rate_limiting_resettime`.

Calls `GET /rate_limit`

Return type `github.RateLimit.RateLimit`

oauth_scopes

Type list of string

get_license (*key=NotSet*)

Calls `GET /license/:license`

Parameters **key** – string

Return type `github.License.License`

get_licenses ()

Calls `GET /licenses`

Return type `github.PaginatedList.PaginatedList` of `github.License.License`

get_user (*login=NotSet*)

Calls `GET /users/:user` or `GET /user`

Parameters **login** – string

Return type `github.NamedUser.NamedUser`

get_users (*since=NotSet*)

Calls `GET /users`

Parameters **since** – integer

Return type `github.PaginatedList.PaginatedList` of `github.NamedUser.NamedUser`

get_organization (*login*)

Calls `GET /orgs/:org`

Parameters **login** – string

Return type `github.Organization.Organization`

get_organizations (*since=NotSet*)

Calls `GET /organizations`

Parameters `since` – integer

Return type `github.PaginatedList.PaginatedList` of `github.Organization.Organization`

get_repo (*full_name_or_id*, *lazy=True*)

Calls `GET /repos/:owner/:repo` or `GET /repositories/:id`

Return type `github.Repository.Repository`

get_repos (*since=NotSet*)

Calls `GET /repositories`

Parameters `since` – integer

Return type `github.PaginatedList.PaginatedList` of `github.Repository.Repository`

get_gist (*id*)

Calls `GET /gists/:id`

Parameters `id` – string

Return type `github.Gist.Gist`

get_gists ()

Calls `GET /gists/public`

Return type `github.PaginatedList.PaginatedList` of `github.Gist.Gist`

search_repositories (*query*, *sort=NotSet*, *order=NotSet*, ***qualifiers*)

Calls `GET /search/repositories`

Parameters

- **query** – string
- **sort** – string ('stars', 'forks', 'updated')
- **order** – string ('asc', 'desc')
- **qualifiers** – keyword dict query qualifiers

Return type `github.PaginatedList.PaginatedList` of `github.Repository.Repository`

search_users (*query*, *sort=NotSet*, *order=NotSet*, ***qualifiers*)

Calls `GET /search/users`

Parameters

- **query** – string
- **sort** – string ('followers', 'repositories', 'joined')
- **order** – string ('asc', 'desc')
- **qualifiers** – keyword dict query qualifiers

Return type `github.PaginatedList.PaginatedList` of `github.NamedUser.NamedUser`

search_issues (*query*, *sort=NotSet*, *order=NotSet*, ***qualifiers*)

Calls `GET /search/issues`

Parameters

- **query** – string
- **sort** – string (‘comments’, ‘created’, ‘updated’)
- **order** – string (‘asc’, ‘desc’)
- **qualifiers** – keyword dict query qualifiers

Return type `github.PaginatedList.PaginatedList` of `github.Issue.Issue`

search_code (*query*, *sort=NotSet*, *order=NotSet*, ***qualifiers*)

Calls `GET /search/code`

Parameters

- **query** – string
- **sort** – string (‘indexed’)
- **order** – string (‘asc’, ‘desc’)
- **qualifiers** – keyword dict query qualifiers

Return type `github.PaginatedList.PaginatedList` of `github.ContentFile.ContentFile`

search_commits (*query*, *sort=NotSet*, *order=NotSet*, ***qualifiers*)

Calls `GET /search/commits`

Parameters

- **query** – string
- **sort** – string (‘author-date’, ‘committer-date’)
- **order** – string (‘asc’, ‘desc’)
- **qualifiers** – keyword dict query qualifiers

Return type `github.PaginatedList.PaginatedList` of `github.Commit.Commit`

render_markdown (*text*, *context=NotSet*)

Calls `POST /markdown`

Parameters

- **text** – string
- **context** – `github.Repository.Repository`

Return type string

get_hook (*name*)

Calls `GET /hooks/:name`

Parameters **name** – string

Return type `github.HookDescription.HookDescription`

get_hooks ()

Calls `GET /hooks`

Return type list of *github.HookDescription.HookDescription*

get_gitignore_templates ()

Calls GET /gitignore/templates

Return type list of string

get_gitignore_template (*name*)

Calls GET /gitignore/templates/:name

Return type *github.GitignoreTemplate.GitignoreTemplate*

get_emojis ()

Calls GET /emojis

Return type dictionary of type => url for emoji‘

create_from_raw_data (*klass*, *raw_data*, *headers*={})

Creates an object from *raw_data* previously obtained by *github.GithubObject.GithubObject.raw_data*, and optionally *headers* previously obtained by *github.GithubObject.GithubObject.raw_headers*.

Parameters

- **klass** – the class of the object to create
- **raw_data** – dict
- **headers** – dict

Return type instance of class *klass*

dump (*obj*, *file*, *protocol*=0)

Dumps (pickles) a PyGithub object to a file-like object. Some effort is made to not pickle sensitive information like the Github credentials used in the *Github* instance. But NO EFFORT is made to remove sensitive information from the object’s attributes.

Parameters

- **obj** – the object to pickle
- **file** – the file-like object to pickle to
- **protocol** – the pickling protocol

load (*f*)

Loads (unpickles) a PyGithub object from a file-like object.

Parameters **f** – the file-like object to unpickle from

Returns the unpickled object

get_api_status ()

This doesn’t work with a Github Enterprise installation, because it always targets <https://status.github.com>.

Calls GET /api/status.json

Return type *github.Status.Status*

get_last_api_status_message ()

This doesn’t work with a Github Enterprise installation, because it always targets <https://status.github.com>.

Calls GET /api/last-message.json

Return type *github.StatusMessage.StatusMessage*

get_api_status_messages ()

This doesn't work with a Github Enterprise installation, because it always targets <https://status.github.com>.

Calls GET /api/messages.json

Return type list of *github.StatusMessage.StatusMessage*

get_installation (id)

Parameters id –

Returns

2.2 APIs

- /api/last-message.json
 - GET: *github.MainClass.Github.get_last_api_status_message ()*
- /api/messages.json
 - GET: *github.MainClass.Github.get_api_status_messages ()*
- /api/status.json
 - GET: *github.MainClass.Github.get_api_status ()*
- /authorizations
 - GET: *github.AuthenticatedUser.AuthenticatedUser.get_authorizations ()*
 - POST: *github.AuthenticatedUser.AuthenticatedUser.create_authorization ()*
- /authorizations/:id
 - GET: *github.AuthenticatedUser.AuthenticatedUser.get_authorization ()*
 - PATCH: *github.Authorization.Authorization.edit ()*
 - DELETE: *github.Authorization.Authorization.delete ()*
- /emojis
 - GET: *github.MainClass.Github.get_emojis ()*
- /events
 - GET: *github.AuthenticatedUser.AuthenticatedUser.get_events ()*
- /gists
 - GET: *github.AuthenticatedUser.AuthenticatedUser.get_gists ()*
 - POST: *github.AuthenticatedUser.AuthenticatedUser.create_gist ()*
- /gists/:gist_id/comments
 - GET: *github.Gist.Gist.get_comments ()*
 - POST: *github.Gist.Gist.create_comment ()*
- /gists/:gist_id/comments/:id
 - GET: *github.Gist.Gist.get_comment ()*
 - PATCH: *github.GistComment.GistComment.edit ()*
 - DELETE: *github.GistComment.GistComment.delete ()*

- /gists/:id
 - GET: `github.MainClass.Github.get_gist()`
 - PATCH: `github.Gist.Gist.edit()`
 - DELETE: `github.Gist.Gist.delete()`
- /gists/:id/forks
 - POST: `github.Gist.Gist.create_fork()`
- /gists/:id/star
 - GET: `github.Gist.Gist.is_starred()`
 - PUT: `github.Gist.Gist.set_starred()`
 - DELETE: `github.Gist.Gist.reset_starred()`
- /gists/public
 - GET: `github.MainClass.Github.get_gists()`
- /gists/starred
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_starred_gists()`
- /gitignore/templates
 - GET: `github.MainClass.Github.get_gitignore_templates()`
- /gitignore/templates/:name
 - GET: `github.MainClass.Github.get_gitignore_template()`
- /hooks
 - GET: `github.MainClass.Github.get_hooks()`
- /hooks/:name
 - GET: `github.MainClass.Github.get_hook()`
- /hub
 - POST: `github.Repository.Repository.subscribe_to_hub()` or `github.Repository.Repository.unsubscribe_from_hub()`
- /installation/repositories
 - GET: `github.Installation.Installation.get_repos()`
- /issues
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_issues()`
- /legacy/issues/search/:owner/:repository/:state/:keyword
 - GET: `github.Repository.Repository.legacy_search_issues()`
- /license/:license
 - GET: `github.MainClass.Github.get_license()`
- /licenses
 - GET: `github.MainClass.Github.get_licenses()`
- /markdown

- POST: `github.MainClass.Github.render_markdown()`
- /markdown/raw
 - POST: Not implemented, see /markdown
- /networks/:owner/:repo/events
 - GET: `github.Repository.Repository.get_network_events()`
- /notifications
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_notifications()`
- /notifications/threads/:id
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_notification()`
- /organizations
 - GET: `github.MainClass.Github.get_organizations()`
- /orgs/:org
 - GET: `github.MainClass.Github.get_organization()`
 - PATCH: `github.Organization.Organization.edit()`
- /orgs/:org/events
 - GET: `github.Organization.Organization.get_events()`
- /orgs/:org/issues
 - GET: `github.Organization.Organization.get_issues()`
- /orgs/:org/members
 - GET: `github.Organization.Organization.get_members()`
- /orgs/:org/members/:user
 - GET: `github.Organization.Organization.has_in_members()`
 - DELETE: `github.Organization.Organization.remove_from_members()`
- /orgs/:org/memberships/:user
 - PUT: `github.Organization.Organization.add_to_members()`
 - DELETE: `github.Organization.Organization.remove_from_membership()`
- /orgs/:org/outside_collaborators
 - GET: `github.Organization.Organization.get_outside_collaborators()`
- /orgs/:org/outside_collaborators/:username
 - PUT: `github.Organization.Organization.convert_to_outside_collaborator()`
 - DELETE: `github.Organization.Organization.remove_outside_collaborator()`
- /orgs/:org/public_members
 - GET: `github.Organization.Organization.get_public_members()`
- /orgs/:org/public_members/:user
 - GET: `github.Organization.Organization.has_in_public_members()`
 - PUT: `github.Organization.Organization.add_to_public_members()`

- DELETE: `github.Organization.Organization.remove_from_public_members()`
- `/orgs/:org/repos`
 - GET: `github.Organization.Organization.get_repos()`
 - POST: `github.Organization.Organization.create_repo()`
- `/orgs/:org/teams`
 - GET: `github.Organization.Organization.get_teams()`
 - POST: `github.Organization.Organization.create_team()`
- `/orgs/:owner/hooks`
 - GET: `github.Organization.Organization.get_hooks()`
 - POST: `github.Organization.Organization.create_hook()`
- `/orgs/:owner/hooks/:id`
 - GET: `github.Organization.Organization.get_hook()`
 - PATCH: `github.Organization.Organization.edit_hook()`
 - DELETE: `github.Organization.Organization.delete_hook()`
- `/rate_limit`
 - GET: Not implemented, see *Github.rate_limiting*
- `/reactions/:id`
 - DELETE: `github.Reaction.Reaction.delete()`
- `/repos/:owner/:repo`
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_repo()` or `github.MainClass.Github.get_repo()` or `github.NamedUser.NamedUser.get_repo()` or `github.Organization.Organization.get_repo()`
 - PATCH: `github.Repository.Repository.edit()`
 - DELETE: `github.Repository.Repository.delete()`
- `/repos/:owner/:repo/:archive_format/:ref`
 - GET: `github.Repository.Repository.get_archive_link()`
- `/repos/:owner/:repo/assignees`
 - GET: `github.Repository.Repository.get_assignees()`
- `/repos/:owner/:repo/assignees/:assignee`
 - GET: `github.Repository.Repository.has_in_assignees()`
- `/repos/:owner/:repo/branches`
 - GET: `github.Repository.Repository.get_branches()`
- `/repos/:owner/:repo/branches/:branch`
 - GET: `github.Repository.Repository.get_branch()` or `github.Repository.Repository.get_protected_branch()`
 - PATCH: `github.Repository.Repository.protect_branch()`
- `/repos/:owner/:repo/collaborators`

- GET: `github.Repository.Repository.get_collaborators()`
- `/repos/:owner/:repo/collaborators/:user`
 - GET: `github.Repository.Repository.has_in_collaborators()`
 - PUT: `github.Repository.Repository.add_to_collaborators()`
 - DELETE: `github.Repository.Repository.remove_from_collaborators()`
- `/repos/:owner/:repo/comments`
 - GET: `github.Repository.Repository.get_comments()`
- `/repos/:owner/:repo/comments/:id`
 - GET: `github.Repository.Repository.get_comment()`
 - PATCH: `github.CommitComment.CommitComment.edit()`
 - DELETE: `github.CommitComment.CommitComment.delete()`
- `/repos/:owner/:repo/comments/:id/reactions`
 - GET: `github.CommitComment.CommitComment.get_reactions()`
 - POST: `github.CommitComment.CommitComment.create_reaction()`
- `/repos/:owner/:repo/commits`
 - GET: `github.Repository.Repository.get_commits()`
- `/repos/:owner/:repo/commits/:ref/status/`
 - GET: `github.Commit.Commit.get_combined_status()`
- `/repos/:owner/:repo/commits/:ref/status``
 - GET: `github.GitRef.GitRef.get_status()`
- `/repos/:owner/:repo/commits/:ref/statuses``
 - GET: `github.GitRef.GitRef.get_statuses()`
- `/repos/:owner/:repo/commits/:sha`
 - GET: `github.Repository.Repository.get_commit()`
- `/repos/:owner/:repo/commits/:sha/comments`
 - GET: `github.Commit.Commit.get_comments()`
 - POST: `github.Commit.Commit.create_comment()`
- `/repos/:owner/:repo/compare/:base...:head`
 - GET: `github.Repository.Repository.compare()`
- `/repos/:owner/:repo/contents/:path`
 - GET: `github.Repository.Repository.get_contents()` or `github.Repository.Repository.get_dir_contents()` or `github.Repository.Repository.get_file_contents()`
 - PUT: `github.Repository.Repository.create_file()` or `github.Repository.Repository.update_file()`
 - DELETE: `github.Repository.Repository.delete_file()`
- `/repos/:owner/:repo/contributors`

- GET: `github.Repository.Repository.get_contributors()`
- `/repos/:owner/:repo/downloads`
 - GET: `github.Repository.Repository.get_downloads()`
- `/repos/:owner/:repo/downloads/:id`
 - GET: `github.Repository.Repository.get_download()`
 - DELETE: `github.Download.Download.delete()`
- `/repos/:owner/:repo/events`
 - GET: `github.Repository.Repository.get_events()`
- `/repos/:owner/:repo/forks`
 - GET: `github.Repository.Repository.get_forks()`
 - POST: `github.AuthenticatedUser.AuthenticatedUser.create_fork()` or `github.Organization.Organization.create_fork()`
- `/repos/:owner/:repo/git/blobs`
 - POST: `github.Repository.Repository.create_git_blob()`
- `/repos/:owner/:repo/git/blobs/:sha`
 - GET: `github.Repository.Repository.get_git_blob()`
- `/repos/:owner/:repo/git/commits`
 - POST: `github.Repository.Repository.create_git_commit()`
- `/repos/:owner/:repo/git/commits/:sha`
 - GET: `github.Repository.Repository.get_git_commit()`
- `/repos/:owner/:repo/git/refs`
 - GET: `github.Repository.Repository.get_git_refs()`
 - POST: `github.Repository.Repository.create_git_ref()`
- `/repos/:owner/:repo/git/refs/:ref`
 - GET: `github.Repository.Repository.get_git_ref()`
 - PATCH: `github.GitRef.GitRef.edit()`
 - DELETE: `github.GitRef.GitRef.delete()`
- `/repos/:owner/:repo/git/tags`
 - POST: `github.Repository.Repository.create_git_tag()`
- `/repos/:owner/:repo/git/tags/:sha`
 - GET: `github.Repository.Repository.get_git_tag()`
- `/repos/:owner/:repo/git/trees`
 - POST: `github.Repository.Repository.create_git_tree()`
- `/repos/:owner/:repo/git/trees/:sha`
 - GET: `github.Repository.Repository.get_git_tree()`
- `/repos/:owner/:repo/hooks`

- GET: `github.Repository.Repository.get_hooks()`
- POST: `github.Repository.Repository.create_hook()`
- `/repos/:owner/:repo/hooks/:id`
 - GET: `github.Repository.Repository.get_hook()`
 - PATCH: `github.Hook.Hook.edit()`
 - DELETE: `github.Hook.Hook.delete()`
- `/repos/:owner/:repo/hooks/:id/pings`
 - POST: `github.Hook.Hook.ping()`
- `/repos/:owner/:repo/hooks/:id/tests`
 - POST: `github.Hook.Hook.test()`
- `/repos/:owner/:repo/import`
 - GET: `github.Repository.Repository.get_source_import()`
 - PUT: `github.Repository.Repository.create_source_import()`
- `/repos/:owner/:repo/issues`
 - GET: `github.Repository.Repository.get_issues()`
 - POST: `github.Repository.Repository.create_issue()`
- `/repos/:owner/:repo/issues/:issue_number/events`
 - GET: `github.Issue.Issue.get_events()`
- `/repos/:owner/:repo/issues/:number`
 - GET: `github.PullRequest.PullRequest.as_issue()` or `github.Repository.Repository.get_issue()`
 - PATCH: `github.Issue.Issue.edit()`
- `/repos/:owner/:repo/issues/:number/assignees`
 - POST: `github.Issue.Issue.add_to_assignees()`
 - DELETE: `github.Issue.Issue.remove_from_assignees()`
- `/repos/:owner/:repo/issues/:number/comments`
 - GET: `github.Issue.Issue.get_comments()` or `github.PullRequest.PullRequest.get_issue_comments()`
 - POST: `github.Issue.Issue.create_comment()` or `github.PullRequest.PullRequest.create_issue_comment()`
- `/repos/:owner/:repo/issues/:number/labels`
 - GET: `github.Issue.Issue.get_labels()` or `github.PullRequest.PullRequest.get_labels()`
 - POST: `github.Issue.Issue.add_to_labels()` or `github.PullRequest.PullRequest.add_to_labels()`
 - PUT: `github.Issue.Issue.set_labels()` or `github.PullRequest.PullRequest.set_labels()`

- DELETE: `github.Issue.Issue.delete_labels()` or `github.PullRequest.PullRequest.delete_labels()`
- /repos/:owner/:repo/issues/:number/labels/:name
 - DELETE: `github.Issue.Issue.remove_from_labels()` or `github.PullRequest.PullRequest.remove_from_labels()`
- /repos/:owner/:repo/issues/:number/reactions
 - GET: `github.Issue.Issue.get_reactions()`
 - POST: `github.Issue.Issue.create_reaction()`
- /repos/:owner/:repo/issues/comments
 - GET: `github.Repository.Repository.get_issues_comments()`
- /repos/:owner/:repo/issues/comments/:id
 - GET: `github.Issue.Issue.get_comment()` or `github.PullRequest.PullRequest.get_issue_comment()`
 - PATCH: `github.IssueComment.IssueComment.edit()`
 - DELETE: `github.IssueComment.IssueComment.delete()`
- /repos/:owner/:repo/issues/comments/:id/reactions
 - GET: `github.IssueComment.IssueComment.get_reactions()`
 - POST: `github.IssueComment.IssueComment.create_reaction()`
- /repos/:owner/:repo/issues/events
 - GET: `github.Repository.Repository.get_issues_events()`
- /repos/:owner/:repo/issues/events/:id
 - GET: `github.Repository.Repository.get_issues_event()`
- /repos/:owner/:repo/keys
 - GET: `github.Repository.Repository.get_keys()`
 - POST: `github.Repository.Repository.create_key()`
- /repos/:owner/:repo/keys/:id
 - GET: `github.Repository.Repository.get_key()`
 - DELETE: `github.RepositoryKey.RepositoryKey.delete()`
- /repos/:owner/:repo/labels
 - GET: `github.Repository.Repository.get_labels()`
 - POST: `github.Repository.Repository.create_label()`
- /repos/:owner/:repo/labels/:name
 - GET: `github.Repository.Repository.get_label()`
 - PATCH: `github.Label.Label.edit()`
 - DELETE: `github.Label.Label.delete()`
- /repos/:owner/:repo/languages
 - GET: `github.Repository.Repository.get_languages()`

- /repos/:owner/:repo/license
 - GET: `github.Repository.Repository.get_license()`
- /repos/:owner/:repo/merges
 - POST: `github.Repository.Repository.merge()`
- /repos/:owner/:repo/milestones
 - GET: `github.Repository.Repository.get_milestones()`
 - POST: `github.Repository.Repository.create_milestone()`
- /repos/:owner/:repo/milestones/:number
 - GET: `github.Repository.Repository.get_milestone()`
 - PATCH: `github.Milestone.Milestone.edit()`
 - DELETE: `github.Milestone.Milestone.delete()`
- /repos/:owner/:repo/milestones/:number/labels
 - GET: `github.Milestone.Milestone.get_labels()`
- /repos/:owner/:repo/pulls
 - GET: `github.Repository.Repository.get_pulls()`
 - POST: `github.Repository.Repository.create_pull()`
- /repos/:owner/:repo/pulls/:number
 - GET: `github.Issue.Issue.as_pull_request()` or `github.Repository.Repository.get_pull()`
 - PATCH: `github.PullRequest.PullRequest.edit()`
- /repos/:owner/:repo/pulls/:number/comments
 - GET: `github.PullRequest.PullRequest.get_comments()` or `github.PullRequest.PullRequest.get_review_comments()`
 - POST: `github.PullRequest.PullRequest.create_comment()` or `github.PullRequest.PullRequest.create_review_comment()`
- /repos/:owner/:repo/pulls/:number/commits
 - GET: `github.PullRequest.PullRequest.get_commits()`
- /repos/:owner/:repo/pulls/:number/files
 - GET: `github.PullRequest.PullRequest.get_files()`
- /repos/:owner/:repo/pulls/:number/merge
 - GET: `github.PullRequest.PullRequest.is_merged()`
 - PUT: `github.PullRequest.PullRequest.merge()`
- /repos/:owner/:repo/pulls/:number/requested_reviewers
 - GET: `github.PullRequest.PullRequest.get_review_requests()`
 - POST: `github.PullRequest.PullRequest.create_review_request()`
 - DELETE: `github.PullRequest.PullRequest.delete_review_request()`
- /repos/:owner/:repo/pulls/:number/review/:id/comments

- GET: `github.PullRequest.PullRequest.get_single_review_comments()`
- `/repos/:owner/:repo/pulls/:number/reviews`
 - GET: `github.PullRequest.PullRequest.get_reviews()`
 - POST: `github.PullRequest.PullRequest.create_review()`
- `/repos/:owner/:repo/pulls/:number/reviews/:id`
 - GET: `github.PullRequest.PullRequest.get_review()`
- `/repos/:owner/:repo/pulls/comments`
 - GET: `github.Repository.Repository.get_pulls_comments()` or `github.Repository.Repository.get_pulls_review_comments()`
- `/repos/:owner/:repo/pulls/comments/:number`
 - GET: `github.PullRequest.PullRequest.get_comment()` or `github.PullRequest.PullRequest.get_review_comment()`
 - PATCH: `github.PullRequestComment.PullRequestComment.edit()`
 - DELETE: `github.PullRequestComment.PullRequestComment.delete()`
- `/repos/:owner/:repo/pulls/comments/:number/reactions`
 - GET: `github.PullRequestComment.PullRequestComment.get_reactions()`
 - POST: `github.PullRequestComment.PullRequestComment.create_reaction()`
- `/repos/:owner/:repo/readme`
 - GET: `github.Repository.Repository.get_readme()`
- `/repos/:owner/:repo/releases`
 - GET: `github.Repository.Repository.get_releases()`
 - POST: `github.Repository.Repository.create_git_release()`
- `/repos/:owner/:repo/releases/:id`
 - GET: `github.Repository.Repository.get_release()`
- `/repos/:owner/:repo/releases/:release_id`
 - PATCH: `github.GitRelease.GitRelease.update_release()`
 - DELETE: `github.GitRelease.GitRelease.delete_release()`
- `/repos/:owner/:repo/releases/:release_id/assets`
 - GET: `github.GitRelease.GitRelease.get_assets()`
- `/repos/:owner/:repo/releases/latest`
 - GET: `github.Repository.Repository.get_latest_release()`
- `/repos/:owner/:repo/stargazers`
 - GET: `github.Repository.Repository.get_stargazers_with_dates()` or `github.Repository.Repository.get_stargazers()`
- `/repos/:owner/:repo/stats/code_frequency`
 - GET: `github.Repository.Repository.get_stats_code_frequency()`
- `/repos/:owner/:repo/stats/commit_activity`

- GET: `github.Repository.Repository.get_stats_commit_activity()`
- `/repos/:owner/:repo/stats/contributors`
 - GET: `github.Repository.Repository.get_stats_contributors()`
- `/repos/:owner/:repo/stats/participation`
 - GET: `github.Repository.Repository.get_stats_participation()`
- `/repos/:owner/:repo/stats/punch_card`
 - GET: `github.Repository.Repository.get_stats_punch_card()`
- `/repos/:owner/:repo/statuses/:ref`
 - GET: `github.Commit.Commit.get_statuses()`
- `/repos/:owner/:repo/statuses/:sha`
 - POST: `github.Commit.Commit.create_status()`
- `/repos/:owner/:repo/subscribers`
 - GET: `github.Repository.Repository.get_subscribers()`
- `/repos/:owner/:repo/subscription`
 - GET: `github.AuthenticatedUser.AuthenticatedUser.has_in_watched()`
 - PUT: `github.AuthenticatedUser.AuthenticatedUser.add_to_watched()`
 - DELETE: `github.AuthenticatedUser.AuthenticatedUser.remove_from_watched()`
- `/repos/:owner/:repo/tags`
 - GET: `github.Repository.Repository.get_tags()`
- `/repos/:owner/:repo/teams`
 - GET: `github.Repository.Repository.get_teams()`
- `/repos/:owner/:repo/topics`
 - GET: `github.Repository.Repository.get_topics()`
 - PUT: `github.Repository.Repository.replace_topics()`
- `/repos/:owner/:repo/watchers`
 - GET: `github.Repository.Repository.get_watchers()`
- `/repositories`
 - GET: `github.MainClass.Github.get_repos()`
- `/repositories/:id`
 - GET: `github.MainClass.Github.get_repo()`
- `/search/code`
 - GET: `github.MainClass.Github.search_code()`
- `/search/commits`
 - GET: `github.MainClass.Github.search_commits()`
- `/search/issues`

- GET: `github.MainClass.Github.search_issues()`
- `/search/repositories`
 - GET: `github.MainClass.Github.search_repositories()`
- `/search/users`
 - GET: `github.MainClass.Github.search_users()`
- `/teams/:id`
 - GET: `github.Organization.Organization.get_team()`
 - PATCH: `github.Team.Team.edit()`
 - DELETE: `github.Team.Team.delete()`
- `/teams/:id/members`
 - GET: `github.Team.Team.get_members()`
- `/teams/:id/members/:user`
 - GET: `github.Team.Team.has_in_members()`
 - PUT: `github.Team.Team.add_to_members()`
 - DELETE: `github.Team.Team.remove_from_members()`
- `/teams/:id/memberships/:user`
 - PUT: `github.Team.Team.add_membership()`
- `/teams/:id/repos`
 - GET: `github.Team.Team.get_repos()`
- `/teams/:id/repos/:org/:repo`
 - PUT: `github.Team.Team.add_to_repos()` or `github.Team.Team.set_repo_permission()`
- `/teams/:id/repos/:owner/:repo`
 - GET: `github.Team.Team.has_in_repos()`
 - DELETE: `github.Team.Team.remove_from_repos()`
- `/teams/:team_id/memberships/:username`
 - DELETE: `github.Team.Team.remove_membership()`
- `/user`
 - GET: `github.MainClass.Github.get_user()`
 - PATCH: `github.AuthenticatedUser.AuthenticatedUser.edit()`
- `/user/emails`
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_emails()`
 - POST: `github.AuthenticatedUser.AuthenticatedUser.add_to_emails()`
 - DELETE: `github.AuthenticatedUser.AuthenticatedUser.remove_from_emails()`
- `/user/followers`
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_followers()`

- /user/following
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_following()`
- /user/following/:user
 - GET: `github.AuthenticatedUser.AuthenticatedUser.has_in_following()`
 - PUT: `github.AuthenticatedUser.AuthenticatedUser.add_to_following()`
 - DELETE: `github.AuthenticatedUser.AuthenticatedUser.remove_from_following()`
- /user/issues
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_user_issues()`
- /user/keys
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_keys()`
 - POST: `github.AuthenticatedUser.AuthenticatedUser.create_key()`
- /user/keys/:id
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_key()`
 - DELETE: `github.UserKey.UserKey.delete()`
- /user/orgs
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_orgs()`
- /user/repos
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_repos()`
 - POST: `github.AuthenticatedUser.AuthenticatedUser.create_repo()`
- /user/repository_invitations/:invitation_id
 - PATCH: `github.AuthenticatedUser.AuthenticatedUser.accept_invitation()`
- /user/starred
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_starred()`
- /user/starred/:owner/:repo
 - GET: `github.AuthenticatedUser.AuthenticatedUser.has_in_starred()`
 - PUT: `github.AuthenticatedUser.AuthenticatedUser.add_to_starred()`
 - DELETE: `github.AuthenticatedUser.AuthenticatedUser.remove_from_starred()`
- /user/subscriptions
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_subscriptions()` or `github.AuthenticatedUser.AuthenticatedUser.get_watched()`
- /user/subscriptions/:owner/:repo
 - GET: `github.AuthenticatedUser.AuthenticatedUser.has_in_subscriptions()`
 - PUT: `github.AuthenticatedUser.AuthenticatedUser.add_to_subscriptions()`
 - DELETE: `github.AuthenticatedUser.AuthenticatedUser.remove_from_subscriptions()`

- /user/teams
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_teams()`
- /users
 - GET: `github.MainClass.Github.get_users()`
- /users/:user
 - GET: `github.MainClass.Github.get_user()`
- /users/:user/events
 - GET: `github.NamedUser.NamedUser.get_events()`
- /users/:user/events/orgs/:org
 - GET: `github.AuthenticatedUser.AuthenticatedUser.get_organization_events()`
- /users/:user/events/public
 - GET: `github.NamedUser.NamedUser.get_public_events()`
- /users/:user/followers
 - GET: `github.NamedUser.NamedUser.get_followers()`
- /users/:user/following
 - GET: `github.NamedUser.NamedUser.get_following()`
- /users/:user/following/:target_user
 - GET: `github.NamedUser.NamedUser.has_in_following()`
- /users/:user/gists
 - GET: `github.NamedUser.NamedUser.get_gists()`
- /users/:user/keys
 - GET: `github.NamedUser.NamedUser.get_keys()`
- /users/:user/orgs
 - GET: `github.NamedUser.NamedUser.get_orgs()`
- /users/:user/received_events
 - GET: `github.NamedUser.NamedUser.get_received_events()`
- /users/:user/received_events/public
 - GET: `github.NamedUser.NamedUser.get_public_received_events()`
- /users/:user/repos
 - GET: `github.NamedUser.NamedUser.get_repos()`
- /users/:user/starred
 - GET: `github.NamedUser.NamedUser.get_starred()`
- /users/:user/subscriptions
 - GET: `github.NamedUser.NamedUser.get_subscriptions()`
- /users/:user/watched
 - GET: `github.NamedUser.NamedUser.get_watched()`

- `https://<upload_url>/repos/:owner/:repo/releases/:release_id/assets?name=foo.zip`
 - POST: `github.GitRelease.GitRelease.upload_asset()`

2.3 Utilities

2.3.1 Logging

`github.enable_console_debug_logging()`

This function sets up a very simple logging configuration (log everything on standard output) that is useful for troubleshooting.

2.3.2 Error Handling

exception `github.GithubException.GithubException(status, data)`

Error handling in PyGithub is done with exceptions. This class is the base of all exceptions raised by PyGithub (but `github.GithubException.BadAttributeException`).

Some other types of exceptions might be raised by underlying libraries, for example for network-related issues.

status

The status returned by the Github API

data

The (decoded) data returned by the Github API

exception `github.GithubException.BadCredentialsException(status, data)`

Exception raised in case of bad credentials (when Github API replies with a 401 or 403 HTML status)

exception `github.GithubException.UnknownObjectException(status, data)`

Exception raised when a non-existing object is requested (when Github API replies with a 404 HTML status)

exception `github.GithubException.BadUserAgentException(status, data)`

Exception raised when request is sent with a bad user agent header (when Github API replies with a 403 bad user agent HTML status)

exception `github.GithubException.RateLimitExceededException(status, data)`

Exception raised when the rate limit is exceeded (when Github API replies with a 403 rate limit exceeded HTML status)

exception `github.GithubException.BadAttributeException(actualValue, expectedType, transformationException)`

Exception raised when Github returns an attribute with the wrong type.

actual_value

The value returned by Github

expected_type

The type PyGithub expected

transformation_exception

The exception raised when PyGithub tried to parse the value

exception `github.GithubException.TwoFactorException(status, data)`

Exception raised when Github requires a onetime password for two-factor authentication

2.3.3 Default argument

`github.NotSet` is a special value for arguments you don't want to provide. You should not have to manipulate it directly, because it's the default value of all parameters accepting it. Just note that it is different from `None`, which is an allowed value for some parameters.

2.3.4 Pagination

class `github.PaginatedList.PaginatedList`

This class abstracts the [pagination of the API](#).

You can simply enumerate through instances of this class:

```
for repo in user.get_repos():
    print(repo.name)
```

If you want to know the total number of items in the list:

```
print(user.get_repos().totalCount)
print(len(user.get_repos()))
```

You can also index them or take slices:

```
second_repo = user.get_repos()[1]
first_repos = user.get_repos()[:10]
```

If you want to iterate in reversed order, just do:

```
for repo in user.get_repos().reversed():
    print(repo.name)
```

And if you really need it, you can explicitly access a specific page:

```
some_repos = user.get_repos().get_page(0)
some_other_repos = user.get_repos().get_page(3)
```

2.3.5 Input classes

class `github.InputFileContent.InputFileContent` (*content, new_name=NotSet*)

This class represents InputFileContents

Parameters

- **content** – string
- **new_name** – string

class `github.InputGitAuthor.InputGitAuthor` (*name, email, date=NotSet*)

This class represents InputGitAuthors

Parameters

- **name** – string
- **email** – string
- **date** – string

```
class github.InputGitTreeElement.InputGitTreeElement (path, mode, type, content=NotSet, sha=NotSet)
```

This class represents InputGitTreeElements

Parameters

- **path** – string
- **mode** – string
- **type** – string
- **content** – string
- **sha** – string

2.4 Github objects

```
class github.GithubObject.GithubObject
```

Base class for all classes representing objects returned by the API.

raw_data

Type dict

raw_headers

Type dict

etag

Type str

last_modified

Type str

get__repr__ (*params*)

Converts the object to a nicely printable string.

2.4.1 AuthenticatedUser

```
class github.AuthenticatedUser.AuthenticatedUser
```

This class represents AuthenticatedUsers as returned by <https://developer.github.com/v3/users/#get-the-authenticated-user>

An AuthenticatedUser object can be created by calling `get_user()` on a Github object.

avatar_url

Type string

bio

Type string

blog

Type string

collaborators

Type integer

company
 Type string

created_at
 Type datetime.datetime

disk_usage
 Type integer

email
 Type string

events_url
 Type string

followers
 Type integer

followers_url
 Type string

following
 Type integer

following_url
 Type string

gists_url
 Type string

gravatar_id
 Type string

hireable
 Type bool

html_url
 Type string

id
 Type integer

location
 Type string

login
 Type string

name
 Type string

organizations_url
 Type string

owned_private_repos

Type integer

plan

Type *github.Plan.Plan*

private_gists

Type integer

public_gists

Type integer

public_repos

Type integer

received_events_url

Type string

repos_url

Type string

site_admin

Type bool

starred_url

Type string

subscriptions_url

Type string

total_private_repos

Type integer

type

Type string

updated_at

Type datetime.datetime

url

Type string

add_to_emails (*emails)

Calls [POST /user/emails](#)

Parameters **email** – string

Return type None

add_to_following (following)

Calls [PUT /user/following/:user](#)

Parameters **following** – *github.NamedUser.NamedUser*

Return type None

add_to_starred (*starred*)

Calls `PUT /user/starred/:owner/:repo`

Parameters **starred** – *github.Repository.Repository*

Return type `None`

add_to_subscriptions (*subscription*)

Calls `PUT /user/subscriptions/:owner/:repo`

Parameters **subscription** – *github.Repository.Repository*

Return type `None`

add_to_watched (*watched*)

Calls `PUT /repos/:owner/:repo/subscription`

Parameters **watched** – *github.Repository.Repository*

Return type `None`

create_authorization (*scopes=NotSet, note=NotSet, note_url=NotSet, client_id=NotSet, client_secret=NotSet, onetime_password=None*)

Calls `POST /authorizations`

Parameters

- **scopes** – list of string
- **note** – string
- **note_url** – string
- **client_id** – string
- **client_secret** – string
- **onetime_password** – string

Return type *github.Authorization.Authorization*

create_fork (*repo*)

Calls `POST /repos/:owner/:repo/forks`

Parameters **repo** – *github.Repository.Repository*

Return type *github.Repository.Repository*

create_gist (*public, files, description=NotSet*)

Calls `POST /gists`

Parameters

- **public** – bool
- **files** – dict of string to *github.InputFileContent.InputFileContent*
- **description** – string

Return type *github.Gist.Gist*

create_key (*title, key*)

Calls `POST /user/keys`

Parameters

- **title** – string
- **key** – string

Return type `github.UserKey.UserKey`

create_repo (*name*, *description=NotSet*, *homepage=NotSet*, *private=NotSet*, *has_issues=NotSet*, *has_wiki=NotSet*, *has_downloads=NotSet*, *has_projects=NotSet*, *auto_init=NotSet*, *license_template=NotSet*, *gitignore_template=NotSet*, *allow_squash_merge=NotSet*, *allow_merge_commit=NotSet*, *allow_rebase_merge=NotSet*)

Calls `POST /user/repos`

Parameters

- **name** – string
- **description** – string
- **homepage** – string
- **private** – bool
- **has_issues** – bool
- **has_wiki** – bool
- **has_downloads** – bool
- **has_projects** – bool
- **auto_init** – bool
- **license_template** – string
- **gitignore_template** – string
- **allow_squash_merge** – bool
- **allow_merge_commit** – bool
- **allow_rebase_merge** – bool

Return type `github.Repository.Repository`

edit (*name=NotSet*, *email=NotSet*, *blog=NotSet*, *company=NotSet*, *location=NotSet*, *hireable=NotSet*, *bio=NotSet*)

Calls `PATCH /user`

Parameters

- **name** – string
- **email** – string
- **blog** – string
- **company** – string
- **location** – string
- **hireable** – bool
- **bio** – string

Return type `None`

get_authorization (*id*)

Calls `GET /authorizations/:id`

Parameters `id` – integer

Return type `github.Authorization.Authorization`

get_authorized_keys()

Calls `GET /authorizations`

Return type `github.PaginatedList.PaginatedList` of `github.Authorization.Authorization`

get_emails()

Calls `GET /user/emails`

Return type list of string

get_events()

Calls `GET /events`

Return type `github.PaginatedList.PaginatedList` of `github.Event.Event`

get_followers()

Calls `GET /user/followers`

Return type `github.PaginatedList.PaginatedList` of `github.NamedUser.NamedUser`

get_following()

Calls `GET /user/following`

Return type `github.PaginatedList.PaginatedList` of `github.NamedUser.NamedUser`

get_gists()

Calls `GET /gists`

Return type `github.PaginatedList.PaginatedList` of `github.Gist.Gist`

get_issues (`filter=NotSet`, `state=NotSet`, `labels=NotSet`, `sort=NotSet`, `direction=NotSet`, `since=NotSet`)

Calls `GET /issues`

Return type `github.PaginatedList.PaginatedList` of `github.Issue.Issue`

Parameters

- **filter** – string
- **state** – string
- **labels** – list of `github.Label.Label`
- **sort** – string
- **direction** – string
- **since** – `datetime.datetime`

Return type `github.PaginatedList.PaginatedList` of `github.Issue.Issue`

get_user_issues (`filter=NotSet`, `state=NotSet`, `labels=NotSet`, `sort=NotSet`, `direction=NotSet`, `since=NotSet`)

Calls `GET /user/issues`

Return type *github.PaginatedList.PaginatedList* of *github.Issue.Issue*

Parameters

- **filter** – string
- **state** – string
- **labels** – list of *github.Label.Label*
- **sort** – string
- **direction** – string
- **since** – datetime.datetime

Return type *github.PaginatedList.PaginatedList* of *github.Issue.Issue*

get_key (*id*)

Calls GET /user/keys/:id

Parameters **id** – integer

Return type *github.UserKey.UserKey*

get_keys ()

Calls GET /user/keys

Return type *github.PaginatedList.PaginatedList* of *github.UserKey.UserKey*

get_notification (*id*)

Calls GET /notifications/threads/:id

Return type *github.Notification.Notification*

get_notifications (*all=NotSet, participating=NotSet*)

Calls GET /notifications

Parameters

- **all** – bool
- **participating** – bool

Return type *github.PaginatedList.PaginatedList* of *github.Notification.Notification*

get_organization_events (*org*)

Calls GET /users/:user/events/orgs/:org

Parameters **org** – *github.Organization.Organization*

Return type *github.PaginatedList.PaginatedList* of *github.Event.Event*

get_orgs ()

Calls GET /user/orgs

Return type *github.PaginatedList.PaginatedList* of *github.Organization.Organization*

get_repo (*name*)

Calls GET /repos/:owner/:repo

Parameters `name` – string

Return type `github.Repository.Repository`

`get_repos` (*type=NotSet, sort=NotSet, direction=NotSet*)

Calls `GET /user/repos`

Parameters

- `type` – string
- `sort` – string
- `direction` – string

Return type `github.PaginatedList.PaginatedList` of `github.Repository.Repository`

`get_starred` ()

Calls `GET /user/starred`

Return type `github.PaginatedList.PaginatedList` of `github.Repository.Repository`

`get_starred_gists` ()

Calls `GET /gists/starred`

Return type `github.PaginatedList.PaginatedList` of `github.Gist.Gist`

`get_subscriptions` ()

Calls `GET /user/subscriptions`

Return type `github.PaginatedList.PaginatedList` of `github.Repository.Repository`

`get_teams` ()

Calls `GET /user/teams`

Return type `github.PaginatedList.PaginatedList` of `github.Team.Team`

`get_watched` ()

Calls `GET /user/subscriptions`

Return type `github.PaginatedList.PaginatedList` of `github.Repository.Repository`

`has_in_following` (*following*)

Calls `GET /user/following/:user`

Parameters `following` – `github.NamedUser.NamedUser`

Return type bool

`has_in_starred` (*starred*)

Calls `GET /user/starred/:owner/:repo`

Parameters `starred` – `github.Repository.Repository`

Return type bool

`has_in_subscriptions` (*subscription*)

Calls `GET /user/subscriptions/:owner/:repo`

Parameters **subscription** – `github.Repository.Repository`

Return type `bool`

has_in_watched (*watched*)

Calls `GET /repos/:owner/:repo/subscription`

Parameters **watched** – `github.Repository.Repository`

Return type `bool`

remove_from_emails (**emails*)

Calls `DELETE /user/emails`

Parameters **email** – `string`

Return type `None`

remove_from_following (*following*)

Calls `DELETE /user/following/:user`

Parameters **following** – `github.NamedUser.NamedUser`

Return type `None`

remove_from_starred (*starred*)

Calls `DELETE /user/starred/:owner/:repo`

Parameters **starred** – `github.Repository.Repository`

Return type `None`

remove_from_subscriptions (*subscription*)

Calls `DELETE /user/subscriptions/:owner/:repo`

Parameters **subscription** – `github.Repository.Repository`

Return type `None`

remove_from_watched (*watched*)

Calls `DELETE /repos/:owner/:repo/subscription`

Parameters **watched** – `github.Repository.Repository`

Return type `None`

accept_invitation (*invitation*)

Calls `PATCH /user/repository_invitations/:invitation_id` <<https://developer.github.com/v3/repos/invitations/>>

Parameters **invitation** – `github.Invitation.Invitation` or `int`

Return type `None`

2.4.2 Authorization

class `github.Authorization.Authorization`

This class represents Authorizations. The reference can be found here https://developer.github.com/v3/oauth_authorized/

app

Type `github.AuthorizationApplication.AuthorizationApplication`

created_at
Type `datetime.datetime`

id
Type `integer`

note
Type `string`

note_url
Type `string`

scopes
Type `list of string`

token
Type `string`

updated_at
Type `datetime.datetime`

url
Type `string`

delete ()
Calls `DELETE /authorizations/:id`
Return type `None`

edit (`scopes=NotSet, add_scopes=NotSet, remove_scopes=NotSet, note=NotSet, note_url=NotSet`)
Calls `PATCH /authorizations/:id`
Parameters

- **scopes** – list of string
- **add_scopes** – list of string
- **remove_scopes** – list of string
- **note** – string
- **note_url** – string

Return type `None`

2.4.3 AuthorizationApplication

class `github.AuthorizationApplication.AuthorizationApplication`
This class represents AuthorizationApplications

name
Type `string`

url

Type string

2.4.4 Branch

class `github.Branch.Branch`

This class represents Branches. The reference can be found here <http://developer.github.com/v3/repos/#list-branches>

commit

Type `github.Commit.Commit`

name

Type string

protected

Type bool

enforcement_level

Type string

contexts

Type list of strings

2.4.5 Commit

class `github.Commit.Commit`

This class represents Commits. The reference can be found here <http://developer.github.com/v3/git/commits/>

author

Type `github.NamedUser.NamedUser`

comments_url

Type string

commit

Type `github.GitCommit.GitCommit`

committer

Type `github.NamedUser.NamedUser`

files

Type list of `github.File.File`

html_url

Type string

parents

Type list of `github.Commit.Commit`

sha

Type string

stats

Type `github.CommitStats.CommitStats`

url

Type string

create_comment (*body*, *line=NotSet*, *path=NotSet*, *position=NotSet*)

Calls POST `/repos/:owner/:repo/commits/:sha/comments`

Parameters

- **body** – string
- **line** – integer
- **path** – string
- **position** – integer

Return type `github.CommitComment.CommitComment`

create_status (*state*, *target_url=NotSet*, *description=NotSet*, *context=NotSet*)

Calls POST `/repos/:owner/:repo/statuses/:sha`

Parameters

- **state** – string
- **target_url** – string
- **description** – string
- **context** – string

Return type `github.CommitStatus.CommitStatus`

get_comments ()

Calls GET `/repos/:owner/:repo/commits/:sha/comments`

Return type `github.PaginatedList.PaginatedList` of `github.CommitComment.CommitComment`

get_statuses ()

Calls GET `/repos/:owner/:repo/statuses/:ref`

Return type `github.PaginatedList.PaginatedList` of `github.CommitStatus.CommitStatus`

get_combined_status ()

Calls GET `/repos/:owner/:repo/commits/:ref/status/`

Return type `github.CommitCombinedStatus.CommitCombinedStatus`

2.4.6 CommitCombinedStatus

class `github.CommitCombinedStatus.CommitCombinedStatus`

This class represents CommitCombinedStatuses. The reference can be found here <https://developer.github.com/v3/repos/statuses/#get-the-combined-status-for-a-specific-ref>

state

Type string

sha
 Type string

total_count
 Type integer

commit_url
 Type string

url
 Type string

repository
 Type *github.Repository.Repository*

statuses
 Type list of CommitStatus

2.4.7 CommitComment

class `github.CommitComment.CommitComment`

This class represents CommitComments. The reference can be found here <https://developer.github.com/v3/repos/comments/>

body
 Type string

commit_id
 Type string

created_at
 Type `datetime.datetime`

html_url
 Type string

id
 Type integer

line
 Type integer

path
 Type string

position
 Type integer

updated_at
 Type `datetime.datetime`

url
 Type string

user
 Type `github.NamedUser.NamedUser`

delete()
 Calls `DELETE /repos/:owner/:repo/comments/:id`
 Return type `None`

edit(*body*)
 Calls `PATCH /repos/:owner/:repo/comments/:id`
 Parameters **body** – string
 Return type `None`

get_reactions()
 Calls `GET /repos/:owner/:repo/comments/:id/reactions`
 Returns
 class `github.PaginatedList.PaginatedList` of `github.Reaction.Reaction`

create_reaction(*reaction_type*)
 Calls `POST /repos/:owner/:repo/comments/:id/reactions`
 Parameters **reaction_type** – string
 Return type `github.Reaction.Reaction`

2.4.8 CommitStats

class `github.CommitStats.CommitStats`

This class represents CommitStatses.

additions

Type `integer`

deletions

Type `integer`

total

Type `integer`

2.4.9 CommitStatus

class `github.CommitStatus.CommitStatus`

This class represents CommitStatuses. The reference can be found here <https://developer.github.com/v3/repos/statuses/>

created_at

Type `datetime.datetime`

creator

Type `github.NamedUser.NamedUser`

description
 Type string

id
 Type integer

state
 Type string

context
 Type string

target_url
 Type string

updated_at
 Type datetime.datetime

url
 Type string

2.4.10 Comparison

class `github.Comparison.Comparison`

This class represents Comparisons

ahead_by
 Type integer

base_commit
 Type *github.Commit.Commit*

behind_by
 Type integer

commits
 Type list of *github.Commit.Commit*

diff_url
 Type string

files
 Type list of *github.File.File*

html_url
 Type string

merge_base_commit
 Type *github.Commit.Commit*

patch_url
 Type string

permalink_url
Type string

status
Type string

total_commits
Type integer

url
Type string

2.4.11 ContentFile

class `github.ContentFile.ContentFile`

This class represents ContentFiles. The reference can be found here <https://developer.github.com/v3/repos/contents/#get-contents>

content
Type string

download_url
Type string

encoding
Type string

git_url
Type string

html_url
Type string

license
Type *github.License.License*

name
Type string

path
Type string

repository
Type *github.Repository.Repository*

sha
Type string

size
Type integer

type
Type string

url
Type string

2.4.12 Download

class github.Download.Download

This class represents Downloads. The reference can be found here <https://developer.github.com/v3/repos/downloads/>

accesskeyid
Type string

acl
Type string

bucket
Type string

content_type
Type string

created_at
Type datetime.datetime

description
Type string

download_count
Type integer

expirationdate
Type datetime.datetime

html_url
Type string

id
Type integer

mime_type
Type string

name
Type string

path
Type string

policy
Type string

prefix
Type string

redirect
 Type bool

s3_url
 Type string

signature
 Type string

size
 Type integer

url
 Type string

delete()
 Calls `DELETE /repos/:owner/:repo/downloads/:id`
 Return type None

2.4.13 Event

class `github.Event.Event`

This class represents Events. The reference can be found here <http://developer.github.com/v3/activity/events/>

actor
 Type `github.NamedUser.NamedUser`

created_at
 Type `datetime.datetime`

id
 Type string

org
 Type `github.Organization.Organization`

payload
 Type dict

public
 Type bool

repo
 Type `github.Repository.Repository`

type
 Type string

2.4.14 File

class `github.File.File`

This class represents Files

additions

Type integer

blob_url

Type string

changes

Type integer

contents_url

Type string

deletions

Type integer

filename

Type string

patch

Type string

previous_filename

Type string

raw_url

Type string

sha

Type string

status

Type string

2.4.15 Gist

class `github.Gist.Gist`

This class represents Gists. The reference can be found here <https://developer.github.com/v3/gists/>

comments

Type integer

comments_url

Type string

commits_url

Type string

created_at

Type datetime.datetime
description
Type string
files
Type dict of string to *github.GistFile.GistFile*
fork_of
Type *github.Gist.Gist*
forks
Type list of *github.Gist.Gist*
forks_url
Type string
git_pull_url
Type string
git_push_url
Type string
history
Type list of *github.GistHistoryState.GistHistoryState*
html_url
Type string
id
Type string
owner
Type *github.NamedUser.NamedUser*
public
Type bool
updated_at
Type datetime.datetime
url
Type string
user
Type *github.NamedUser.NamedUser*
create_comment (*body*)
Calls POST /gists/:gist_id/comments
Parameters **body** – string
Return type *github.GistComment.GistComment*
create_fork ()

Calls `POST /gists/:id/forks`

Return type `github.Gist.Gist`

delete ()

Calls `DELETE /gists/:id`

Return type `None`

edit (*description=NotSet, files=NotSet*)

Calls `PATCH /gists/:id`

Parameters

- **description** – string
- **files** – dict of string to `github.InputFileContent.InputFileContent`

Return type `None`

get_comment (*id*)

Calls `GET /gists/:gist_id/comments/:id`

Parameters **id** – integer

Return type `github.GistComment.GistComment`

get_comments ()

Calls `GET /gists/:gist_id/comments`

Return type `github.PaginatedList.PaginatedList` of `github.GistComment.GistComment`

is_starred ()

Calls `GET /gists/:id/star`

Return type `bool`

reset_starred ()

Calls `DELETE /gists/:id/star`

Return type `None`

set_starred ()

Calls `PUT /gists/:id/star`

Return type `None`

2.4.16 GistComment

class `github.GistComment.GistComment`

This class represents GistComments. The reference can be found here <https://developer.github.com/v3/gists/comments/>

body

Type `string`

created_at

Type `datetime.datetime`

id
 Type integer

updated_at
 Type datetime.datetime

url
 Type string

user
 Type *github.NamedUser.NamedUser*

delete()
 Calls DELETE /gists/:gist_id/comments/:id
 Return type None

edit(*body*)
 Calls PATCH /gists/:gist_id/comments/:id
 Parameters **body** – string
 Return type None

2.4.17 GistFile

class `github.GistFile.GistFile`

This class represents GistFiles

content
 Type string

filename
 Type string

language
 Type string

raw_url
 Type string

size
 Type integer

type
 Type string

2.4.18 GistHistoryState

class `github.GistHistoryState.GistHistoryState`

This class represents GistHistoryStates

change_status

Type *github.CommitStats.CommitStats*

comments
Type integer

comments_url
Type string

commits_url
Type string

committed_at
Type datetime.datetime

created_at
Type datetime.datetime

description
Type string

files
Type dict of string to *github.GistFile.GistFile*

forks
Type list of *github.Gist.Gist*

forks_url
Type string

git_pull_url
Type string

git_push_url
Type string

history
Type list of *GistHistoryState*

html_url
Type string

id
Type string

owner
Type *github.NamedUser.NamedUser*

public
Type bool

updated_at
Type datetime.datetime

url

Type string

user

Type *github.NamedUser.NamedUser*

version

Type string

2.4.19 GitAuthor

class `github.GitAuthor.GitAuthor`

This class represents GitAuthors

date

Type `datetime.datetime`

email

Type string

name

Type string

2.4.20 GitBlob

class `github.GitBlob.GitBlob`

This class represents GitBlobs. The reference can be found here <https://developer.github.com/v3/git/blobs/>

content

Type string

encoding

Type string

sha

Type string

size

Type integer

url

Type string

2.4.21 GitCommit

class `github.GitCommit.GitCommit`

This class represents GitCommits. The reference can be found here <https://developer.github.com/v3/git/commits/>

author

Type *github.GitAuthor.GitAuthor*

committer

Type *github.GitAuthor.GitAuthor*

html_url

Type string

message

Type string

parents

Type list of *github.GitCommit.GitCommit*

sha

Type string

tree

Type *github.GitTree.GitTree*

url

Type string

2.4.22 GitObject

class `github.GitObject.GitObject`

This class represents GitObjects

sha

Type string

type

Type string

url

Type string

2.4.23 GitRef

class `github.GitRef.GitRef`

This class represents GitRefs. The reference can be found here <https://developer.github.com/v3/git/refs/>

object

Type *github.GitObject.GitObject*

ref

Type string

url

Type string

delete()

Calls `DELETE /repos/:owner/:repo/git/refs/:ref`

Return type None

edit (*sha*, *force=NotSet*)

Calls `PATCH /repos/:owner/:repo/git/refs/:ref`

Parameters

- **sha** – string
- **force** – bool

Return type None

get_statuses ()

`https://developer.github.com/v3/repos/statuses/#list-statuses-for-a-specific-ref` :calls: `GET /repos/:owner/:repo/commits/:ref/statuses` :return:

get_status ()

`https://developer.github.com/v3/repos/statuses/#get-the-combined-status-for-a-specific-ref` :calls: `GET /repos/:owner/:repo/commits/:ref/status` :return:

2.4.24 GitRelease

class `github.GitRelease.GitRelease`

This class represents GitReleases. The reference can be found here <https://developer.github.com/v3/repos/releases>

id

Type integer

body

Type string

title

Type string

tag_name

Type string

target_commitish

Type string

draft

Type bool

prerelease

Type bool

author

Type `github.GitAuthor.GitAuthor`

created_at

Type `datetime.datetime`

published_at

Type `datetime.datetime`

url

upload_url
Type string

html_url
Type string

tarball_url
Type string

zipball_url
Type string

delete_release()
Calls DELETE /repos/:owner/:repo/releases/:release_id
Return type None

update_release(name, message, draft=False, prerelease=False, tag_name=NotSet, target_commitish=NotSet)
Calls PATCH /repos/:owner/:repo/releases/:release_id
Return type *github.GitRelease.GitRelease*

upload_asset(path, label="", content_type="")
Calls POST https://<upload_url>/repos/:owner/:repo/releases/:release_id/assets?name=foo.zip
Return type *github.GitReleaseAsset.GitReleaseAsset*

get_assets()
Calls GET /repos/:owner/:repo/releases/:release_id/assets
Return type *github.PaginatedList.PaginatedList*

2.4.25 GitReleaseAsset

class *github.GitReleaseAsset.GitReleaseAsset*

This class represents GitReleaseAssets. The reference can be found here <https://developer.github.com/v3/repos/releases/#get-a-single-release-asset>

url
Type string

id
Type integer

name
Type string

label
Type string

content_type
Type string

state
 Type string
size
 Type integer
download_count
 Type integer
created_at
 Type datetime
updated_at
 Type datetime
browser_download_url
 Type string
uploader
 Type github.NamedUser.NamedUser
delete_asset ()
 Delete asset from the release. :rtype: bool
update_asset (*name*, *label=""*)
 Update asset metadata. :rtype: github.GitReleaseAsset.GitReleaseAsset

2.4.26 GitTag

class github.GitTag.**GitTag**

This class represents GitTags. The reference can be found here <https://developer.github.com/v3/git/tags/>

message
 Type string
object
 Type *github.GitObject.GitObject*
sha
 Type string
tag
 Type string
tagger
 Type *github.GitAuthor.GitAuthor*
url
 Type string

2.4.27 GitTree

class `github.GitTree.GitTree`

This class represents GitTrees. The reference can be found here <https://developer.github.com/v3/git/trees/>

sha

Type string

tree

Type list of `github.GitTreeElement.GitTreeElement`

url

Type string

2.4.28 GitTreeElement

class `github.GitTreeElement.GitTreeElement`

This class represents GitTreeElements

mode

Type string

path

Type string

sha

Type string

size

Type integer

type

Type string

url

Type string

2.4.29 GitignoreTemplate

class `github.GitignoreTemplate.GitignoreTemplate`

This class represents GitignoreTemplates. The reference can be found here <https://developer.github.com/v3/gitignore/>

source

Type string

name

Type string

2.4.30 Hook

class `github.Hook.Hook`

This class represents Hooks. The reference can be found here <http://developer.github.com/v3/repos/hooks>

active

Type `bool`

config

Type `dict`

created_at

Type `datetime.datetime`

events

Type `list of string`

id

Type `integer`

last_response

Type `github.HookResponse.HookResponse`

name

Type `string`

test_url

Type `string`

updated_at

Type `datetime.datetime`

url

Type `string`

ping_url

Type `string`

delete()

Calls `DELETE /repos/:owner/:repo/hooks/:id`

Return type `None`

edit (*name*, *config*, *events=NotSet*, *add_events=NotSet*, *remove_events=NotSet*, *active=NotSet*)

Calls `PATCH /repos/:owner/:repo/hooks/:id`

Parameters

- **name** – string
- **config** – dict
- **events** – list of string
- **add_events** – list of string
- **remove_events** – list of string

- **active** – bool

Return type None

test ()

Calls `POST /repos/:owner/:repo/hooks/:id/tests`

Return type None

ping ()

Calls `POST /repos/:owner/:repo/hooks/:id/pings`

Return type None

2.4.31 HookDescription

class `github.HookDescription.HookDescription`

This class represents HookDescriptions

events

Type list of string

name

Type string

schema

Type list of list of string

supported_events

Type list of string

2.4.32 HookResponse

class `github.HookResponse.HookResponse`

This class represents HookResponses

code

Type integer

message

Type string

status

Type string

2.4.33 Installation

class `github.Installation.Installation`

This class represents Installations. The reference can be found here <https://developer.github.com/v3/apps/installations/>

get_repos ()

Calls `GET /installation/repositories`

Return type *github.PaginatedList.PaginatedList* of *github.Repository.Repository*

2.4.34 InstallationAuthorization

class `github.InstallationAuthorization.InstallationAuthorization`

This class represents InstallationAuthorizations

token

Type string

expires_at

Type datetime

on_behalf_of

Type *github.NamedUser.NamedUser*

2.4.35 Invitation

class `github.Invitation.Invitation`

This class represents repository invitations. The reference can be found here <https://developer.github.com/v3/repos/invitations/>

id

Type integer

permissions

Type string

created_at

Type string

url

Type string

html_url

Type string

repository

Type Repository

2.4.36 Issue

class `github.Issue.Issue`

This class represents Issues. The reference can be found here <https://developer.github.com/v3/issues/>

assignee

Type *github.NamedUser.NamedUser*

assignees

Type list of *github.NamedUser.NamedUser*

body
Type string

closed_at
Type datetime.datetime

closed_by
Type *github.NamedUser.NamedUser*

comments
Type integer

comments_url
Type string

created_at
Type datetime.datetime

events_url
Type string

html_url
Type string

id
Type integer

labels
Type list of *github.Label.Label*

labels_url
Type string

milestone
Type *github.Milestone.Milestone*

number
Type integer

pull_request
Type *github.IssuePullRequest.IssuePullRequest*

repository
Type *github.Repository.Repository*

state
Type string

title
Type string

updated_at
Type datetime.datetime

url
 Type string

user
 Type *github.NamedUser.NamedUser*

as_pull_request ()
 Calls GET /repos/:owner/:repo/pulls/:number
 Return type *github.PullRequest.PullRequest*

add_to_assignees (*assignees)
 Calls POST /repos/:owner/:repo/issues/:number/assignees
 Parameters **assignee** – *github.NamedUser.NamedUser* or string
 Return type None

add_to_labels (*labels)
 Calls POST /repos/:owner/:repo/issues/:number/labels
 Parameters **label** – *github.Label.Label* or string
 Return type None

create_comment (body)
 Calls POST /repos/:owner/:repo/issues/:number/comments
 Parameters **body** – string
 Return type *github.IssueComment.IssueComment*

delete_labels ()
 Calls DELETE /repos/:owner/:repo/issues/:number/labels
 Return type None

edit (title=NotSet, body=NotSet, assignee=NotSet, state=NotSet, milestone=NotSet, labels=NotSet, assignees=NotSet)
 Calls PATCH /repos/:owner/:repo/issues/:number
 Parameters

- **title** – string
- **body** – string
- **assignee** – string or *github.NamedUser.NamedUser* or None
- **assignees** – list (of string or *github.NamedUser.NamedUser*)
- **state** – string
- **milestone** – *github.Milestone.Milestone* or None
- **labels** – list of string

Return type None

get_comment (id)
 Calls GET /repos/:owner/:repo/issues/comments/:id
 Parameters **id** – integer

Return type `github.IssueComment.IssueComment`

get_comments (*since=NotSet*)

Calls GET /repos/:owner/:repo/issues/:number/comments

Parameters **since** – datetime.datetime format YYYY-MM-DDTHH:MM:SSZ

Return type `github.PaginatedList.PaginatedList` of `github.IssueComment.IssueComment`

get_events ()

Calls GET /repos/:owner/:repo/issues/:issue_number/events

Return type `github.PaginatedList.PaginatedList` of `github.IssueEvent.IssueEvent`

get_labels ()

Calls GET /repos/:owner/:repo/issues/:number/labels

Return type `github.PaginatedList.PaginatedList` of `github.Label.Label`

remove_from_assignees (**assignees*)

Calls DELETE /repos/:owner/:repo/issues/:number/assignees

Parameters **assignee** – `github.NamedUser.NamedUser` or string

Return type None

remove_from_labels (*label*)

Calls DELETE /repos/:owner/:repo/issues/:number/labels/:name

Parameters **label** – `github.Label.Label` or string

Return type None

set_labels (**labels*)

Calls PUT /repos/:owner/:repo/issues/:number/labels

Parameters **labels** – list of `github.Label.Label` or strings

Return type None

get_reactions ()

Calls GET /repos/:owner/:repo/issues/:number/reactions

Returns

`class github.PaginatedList.PaginatedList` of `github.Reaction.Reaction`

create_reaction (*reaction_type*)

Calls POST /repos/:owner/:repo/issues/:number/reactions

Parameters **reaction_type** – string

Return type `github.Reaction.Reaction`

2.4.37 IssueComment

class `github.IssueComment.IssueComment`

This class represents IssueComments. The reference can be found here <https://developer.github.com/v3/issues/comments/>

body

Type string

created_at

Type datetime.datetime

id

Type integer

issue_url

Type string

updated_at

Type datetime.datetime

url

Type string

html_url

Type string

user

Type `github.NamedUser.NamedUser`

delete()

Calls `DELETE /repos/:owner/:repo/issues/comments/:id`

Return type None

edit(*body*)

Calls `PATCH /repos/:owner/:repo/issues/comments/:id`

Parameters **body** – string

Return type None

get_reactions()

Calls `GET /repos/:owner/:repo/issues/comments/:id/reactions`

Returns

`class github.PaginatedList.PaginatedList` of `github.Reaction.Reaction`

create_reaction(*reaction_type*)

Calls `POST /repos/:owner/:repo/issues/comments/:id/reactions`

Parameters **reaction_type** – string

Return type `github.Reaction.Reaction`

2.4.38 IssueEvent

class `github.IssueEvent.IssueEvent`

This class represents IssueEvents. The reference can be found here <https://developer.github.com/v3/issues/events/>

actor

Type `github.NamedUser.NamedUser`

commit_id

Type string

created_at

Type `datetime.datetime`

event

Type string

id

Type integer

issue

Type `github.Issue.Issue`

url

Type string

2.4.39 IssuePullRequest

class `github.IssuePullRequest.IssuePullRequest`

This class represents IssuePullRequests

diff_url

Type string

html_url

Type string

patch_url

Type string

2.4.40 Label

class `github.Label.Label`

This class represents Labels. The reference can be found here <http://developer.github.com/v3/issues/labels/>

color

Type string

description

Type string

name

url
 Type string

delete()
 Calls `DELETE /repos/:owner/:repo/labels/:name`
 Return type None

edit (*name, color, description=NotSet*)
 Calls `PATCH /repos/:owner/:repo/labels/:name`
 Parameters

- **name** – string
- **color** – string
- **description** – string

Return type None

2.4.41 License

class `github.License.License`

This class represents Licenses. The reference can be found here <https://developer.github.com/v3/licenses/>

key
 Type string

name
 Type string

spdx_id
 Type string

url
 Type string

html_url
 Type string

description
 Type string

implementation
 Type string

body
 Type string

permissions
 Type list of string

conditions

Type list of string

limitations

Type list of string

2.4.42 Milestone

class `github.Milestone.Milestone`

This class represents Milestones. The reference can be found here <http://developer.github.com/v3/issues/milestones/>

closed_issues

Type integer

created_at

Type `datetime.datetime`

creator

Type `github.NamedUser.NamedUser`

description

Type string

due_on

Type `datetime.datetime`

id

Type integer

labels_url

Type string

number

Type integer

open_issues

Type integer

state

Type string

title

Type string

updated_at

Type `datetime.datetime`

url

Type string

delete()

Calls `DELETE /repos/:owner/:repo/milestones/:number`

Return type None

edit (*title*, *state=NotSet*, *description=NotSet*, *due_on=NotSet*)

Calls `PATCH /repos/:owner/:repo/milestones/:number`

Parameters

- **title** – string
- **state** – string
- **description** – string
- **due_on** – date

Return type None

get_labels ()

Calls `GET /repos/:owner/:repo/milestones/:number/labels`

Return type `github.PaginatedList.PaginatedList` of `github.Label.Label`

2.4.43 NamedUser

class `github.NamedUser.NamedUser`

This class represents NamedUsers. The reference can be found here <https://developer.github.com/v3/users/#get-a-single-user>

avatar_url

Type string

bio

Type string

blog

Type string

collaborators

Type integer

company

Type string

created_at

Type `datetime.datetime`

disk_usage

Type integer

email

Type string

events_url

Type string

followers

Type integer

followers_url

Type string
following
Type integer
following_url
Type string
gists_url
Type string
gravatar_id
Type string
hireable
Type bool
html_url
Type string
id
Type integer
location
Type string
login
Type string
name
Type string
organizations_url
Type string
owned_private_repos
Type integer
permissions
Type *github.Permissions.Permissions*
plan
Type *github.Plan.Plan*
private_gists
Type integer
public_gists
Type integer
public_repos
Type integer
received_events_url

Type string
repos_url
Type string
site_admin
Type bool
starred_url
Type string
subscriptions_url
Type string
total_private_repos
Type integer
type
Type string
updated_at
Type datetime.datetime
url
Type string
get_events()
Calls `GET /users/:user/events`
Return type `github.PaginatedList.PaginatedList` of `github.Event.Event`
get_followers()
Calls `GET /users/:user/followers`
Return type `github.PaginatedList.PaginatedList` of `github.NamedUser.NamedUser`
get_following()
Calls `GET /users/:user/following`
Return type `github.PaginatedList.PaginatedList` of `github.NamedUser.NamedUser`
get_gists()
Calls `GET /users/:user/gists`
Return type `github.PaginatedList.PaginatedList` of `github.Gist.Gist`
get_keys()
Calls `GET /users/:user/keys`
Return type `github.PaginatedList.PaginatedList` of `github.UserKey.UserKey`
get_orgs()
Calls `GET /users/:user/orgs`

Return type *github.PaginatedList.PaginatedList* of *github.Organization.Organization*

get_public_events ()

Calls `GET /users/:user/events/public`

Return type *github.PaginatedList.PaginatedList* of *github.Event.Event*

get_public_received_events ()

Calls `GET /users/:user/received_events/public`

Return type *github.PaginatedList.PaginatedList* of *github.Event.Event*

get_received_events ()

Calls `GET /users/:user/received_events`

Return type *github.PaginatedList.PaginatedList* of *github.Event.Event*

get_repo (*name*)

Calls `GET /repos/:owner/:repo`

Parameters **name** – string

Return type *github.Repository.Repository*

get_repos (*type=NotSet, sort=NotSet, direction=NotSet*)

Calls `GET /users/:user/repos`

Parameters

- **type** – string
- **sort** – string
- **direction** – string

Return type *github.PaginatedList.PaginatedList* of *github.Repository.Repository*

get_starred ()

Calls `GET /users/:user/starred`

Return type *github.PaginatedList.PaginatedList* of *github.Repository.Repository*

get_subscriptions ()

Calls `GET /users/:user/subscriptions`

Return type *github.PaginatedList.PaginatedList* of *github.Repository.Repository*

get_watched ()

Calls `GET /users/:user/watched`

Return type *github.PaginatedList.PaginatedList* of *github.Repository.Repository*

has_in_following (*following*)

Calls `GET /users/:user/following/:target_user`

Parameters **following** – *github.NamedUser.NamedUser*

Return type bool

2.4.44 Notification

class `github.Notification.Notification`

This class represents Notifications. The reference can be found here <http://developer.github.com/v3/activity/notifications/>

id

Type string

last_read_at

Type `datetime.datetime`

repository

Type `github.Repository.Repository`

subject

Type `github.NotificationSubject.NotificationSubject`

reason

Type string

subscription_url

Type string

unread

Type bool

updated_at

Type `datetime.datetime`

url

Type string

2.4.45 NotificationSubject

class `github.NotificationSubject.NotificationSubject`

This class represents Subjects of Notifications. The reference can be found here <http://developer.github.com/v3/activity/notifications/#list-your-notifications>

title

Type string

url

Type string

latest_comment_url

Type string

type

Type string

2.4.46 Organization

class `github.Organization.Organization`

This class represents Organizations. The reference can be found here <http://developer.github.com/v3/orgs/>

avatar_url

Type string

billing_email

Type string

blog

Type string

collaborators

Type integer

company

Type string

created_at

Type datetime.datetime

description

Type string

disk_usage

Type integer

email

Type string

events_url

Type string

followers

Type integer

following

Type integer

gravatar_id

Type string

html_url

Type string

id

Type integer

location

Type string

login

Type string
members_url
Type string
name
Type string
owned_private_repos
Type integer
plan
Type *github.Plan.Plan*
private_gists
Type integer
public_gists
Type integer
public_members_url
Type string
public_repos
Type integer
repos_url
Type string
total_private_repos
Type integer
type
Type string
updated_at
Type datetime.datetime
url
Type string
add_to_members (*member*, *role=NotSet*)
Calls `PUT /orgs/:org/memberships/:user`
Parameters

- **member** – *github.NamedUser.NamedUser*
- **role** – string

Return type None
add_to_public_members (*public_member*)
Calls `PUT /orgs/:org/public_members/:user`
Parameters **public_member** – *github.NamedUser.NamedUser*

Return type None

create_fork (*repo*)

Calls POST /repos/:owner/:repo/forks

Parameters **repo** – *github.Repository.Repository*

Return type *github.Repository.Repository*

create_hook (*name, config, events=NotSet, active=NotSet*)

Calls POST /orgs/:owner/hooks

Parameters

- **name** – string
- **config** – dict
- **events** – list of string
- **active** – bool

Return type *github.Hook.Hook*

create_repo (*name, description=NotSet, homepage=NotSet, private=NotSet, has_issues=NotSet, has_wiki=NotSet, has_downloads=NotSet, has_projects=NotSet, team_id=NotSet, auto_init=NotSet, license_template=NotSet, gitignore_template=NotSet, allow_squash_merge=NotSet, allow_merge_commit=NotSet, allow_rebase_merge=NotSet*)

Calls POST /orgs/:org/repos

Parameters

- **name** – string
- **description** – string
- **homepage** – string
- **private** – bool
- **has_issues** – bool
- **has_wiki** – bool
- **has_downloads** – bool
- **has_projects** – bool
- **team_id** – : int
- **auto_init** – bool
- **license_template** – string
- **gitignore_template** – string
- **allow_squash_merge** – bool
- **allow_merge_commit** – bool
- **allow_rebase_merge** – bool

Return type *github.Repository.Repository*

create_team (*name, repo_names=NotSet, permission=NotSet, privacy=NotSet*)

Calls POST /orgs/:org/teams

Parameters

- **name** – string
- **repo_names** – list of *github.Repository.Repository*
- **permission** – string
- **privacy** – string

Return type *github.Team.Team*

delete_hook (*id*)

Calls DELETE /orgs/:owner/hooks/:id

Parameters **id** – integer

Return type None‘

edit (*billing_email=NotSet, blog=NotSet, company=NotSet, description=NotSet, email=NotSet, location=NotSet, name=NotSet*)

Calls PATCH /orgs/:org

Parameters

- **billing_email** – string
- **blog** – string
- **company** – string
- **description** – string
- **email** – string
- **location** – string
- **name** – string

Return type None

edit_hook (*id, name, config, events=NotSet, active=NotSet*)

Calls PATCH /orgs/:owner/hooks/:id

Parameters

- **id** – integer
- **name** – string
- **config** – dict
- **events** – list of string
- **active** – bool

Return type *github.Hook.Hook*

get_events ()

Calls GET /orgs/:org/events

Return type *github.PaginatedList.PaginatedList* of *github.Event.Event*

get_hook (*id*)

Calls GET /orgs/:owner/hooks/:id

Parameters **id** – integer

Return type *github.Hook.Hook*

get_hooks ()

Calls GET /orgs/:owner/hooks

Return type *github.PaginatedList.PaginatedList* of *github.Hook.Hook*

get_issues (*filter=NotSet, state=NotSet, labels=NotSet, sort=NotSet, direction=NotSet, since=NotSet*)

Calls GET /orgs/:org/issues

Return type *github.PaginatedList.PaginatedList* of *github.Issue.Issue*

Parameters

- **filter** – string
- **state** – string
- **labels** – list of *github.Label.Label*
- **sort** – string
- **direction** – string
- **since** – *datetime.datetime*

Return type *github.PaginatedList.PaginatedList* of *github.Issue.Issue*

get_members (*filter_=NotSet, role=NotSet*)

Calls GET /orgs/:org/members

Parameters

- **filter** – string
- **role** – string

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

get_public_members ()

Calls GET /orgs/:org/public_members

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

get_outside_collaborators (*filter_=NotSet*)

Calls GET /orgs/:org/outside_collaborators

Parameters **filter** – string

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

remove_outside_collaborator (*collaborator*)

Calls DELETE /orgs/:org/outside_collaborators/:username

Parameters **collaborator** – *github.NamedUser.NamedUser*

Return type None

convert_to_outside_collaborator (*member*)

Calls PUT /orgs/:org/outside_collaborators/:username

Parameters member – *github.NamedUser.NamedUser*

Return type None

get_repo (*name*)

Calls GET /repos/:owner:repo

Parameters name – string

Return type *github.Repository.Repository*

get_repos (*type=NotSet*)

Calls GET /orgs/:org/repos

Parameters type – string ('all', 'public', 'private', 'forks', 'sources', 'member')

Return type *github.PaginatedList.PaginatedList* of *github.Repository.Repository*

get_team (*id*)

Calls GET /teams/:id

Parameters id – integer

Return type *github.Team.Team*

get_teams ()

Calls GET /orgs/:org/teams

Return type *github.PaginatedList.PaginatedList* of *github.Team.Team*

has_in_members (*member*)

Calls GET /orgs/:org/members/:user

Parameters member – *github.NamedUser.NamedUser*

Return type bool

has_in_public_members (*public_member*)

Calls GET /orgs/:org/public_members/:user

Parameters public_member – *github.NamedUser.NamedUser*

Return type bool

remove_from_membership (*member*)

Calls DELETE /orgs/:org/memberships/:user

Parameters member – *github.NamedUser.NamedUser*

Return type None

remove_from_members (*member*)

Calls DELETE /orgs/:org/members/:user

Parameters member – *github.NamedUser.NamedUser*

Return type None

remove_from_public_members (*public_member*)

Calls DELETE /orgs/:org/public_members/:user

Parameters **public_member** – *github.NamedUser.NamedUser*

Return type None

2.4.47 Permissions

class `github.Permissions.Permissions`

This class represents Permissions

admin

Type bool

pull

Type bool

push

Type bool

2.4.48 Plan

class `github.Plan.Plan`

This class represents Plans

collaborators

Type integer

name

Type string

private_repos

Type integer

space

Type integer

2.4.49 PullRequest

class `github.PullRequest.PullRequest`

This class represents PullRequests. The reference can be found here <http://developer.github.com/v3/pulls/>

additions

Type integer

assignee

Type *github.NamedUser.NamedUser*

assignees

Type list of *github.NamedUser.NamedUser*

base

Type *github.PullRequestPart.PullRequestPart*

body
Type string

changed_files
Type integer

closed_at
Type datetime.datetime

comments
Type integer

comments_url
Type string

commits
Type integer

commits_url
Type string

created_at
Type datetime.datetime

deletions
Type integer

diff_url
Type string

head
Type *github.PullRequestPart.PullRequestPart*

html_url
Type string

id
Type integer

issue_url
Type string

labels
Type list of *github.Label.Label*

merge_commit_sha
Type string

mergeable
Type bool

mergeable_state
Type string

merged

Type bool

merged_at

Type datetime.datetime

merged_by

Type *github.NamedUser.NamedUser*

milestone

Type *github.Milestone.Milestone*

number

Type integer

patch_url

Type string

review_comment_url

Type string

review_comments

Type integer

review_comments_url

Type string

state

Type string

title

Type string

updated_at

Type datetime.datetime

url

Type string

user

Type *github.NamedUser.NamedUser*

as_issue()

Calls `GET /repos/:owner/:repo/issues/:number`

Return type *github.Issue.Issue*

create_comment (*body, commit_id, path, position*)

Calls `POST /repos/:owner/:repo/pulls/:number/comments`

Parameters

- **body** – string
- **commit_id** – *github.Commit.Commit*

- **path** – string
- **position** – integer

Return type *github.PullRequestComment.PullRequestComment*

create_review_comment (*body, commit_id, path, position*)

Calls POST /repos/:owner/:repo/pulls/:number/comments

Parameters

- **body** – string
- **commit_id** – *github.Commit.Commit*
- **path** – string
- **position** – integer

Return type *github.PullRequestComment.PullRequestComment*

create_issue_comment (*body*)

Calls POST /repos/:owner/:repo/issues/:number/comments

Parameters **body** – string

Return type *github.IssueComment.IssueComment*

create_review (*commit, body, event=NotSet, comments=NotSet*)

Calls POST /repos/:owner/:repo/pulls/:number/reviews

Parameters

- **commit** – *github.Commit.Commit*
- **body** – string
- **event** – string
- **comments** – list

Return type *github.PullRequestReview.PullRequestReview*

create_review_request (*reviewers=NotSet, team_reviewers=NotSet*)

Calls POST /repos/:owner/:repo/pulls/:number/requested_reviewers

Parameters

- **reviewers** – list of strings
- **team_reviewers** – list of strings

Return type None

delete_review_request (*reviewers=NotSet, team_reviewers=NotSet*)

Calls DELETE /repos/:owner/:repo/pulls/:number/requested_reviewers

Parameters

- **reviewers** – list of strings
- **team_reviewers** – list of strings

Return type None

edit (*title=NotSet, body=NotSet, state=NotSet, base=NotSet*)

Calls `PATCH /repos/:owner/:repo/pulls/:number`

Parameters

- **title** – string
- **body** – string
- **state** – string
- **base** – string

Return type None

get_comment (*id*)

Calls `GET /repos/:owner/:repo/pulls/comments/:number`

Parameters **id** – integer

Return type `github.PullRequestComment.PullRequestComment`

get_review_comment (*id*)

Calls `GET /repos/:owner/:repo/pulls/comments/:number`

Parameters **id** – integer

Return type `github.PullRequestComment.PullRequestComment`

get_comments ()

Warning: this only returns review comments. For normal conversation comments, use `get_issue_comments`.

Calls `GET /repos/:owner/:repo/pulls/:number/comments`

Return type `github.PaginatedList.PaginatedList` of `github.PullRequestComment.PullRequestComment`

get_review_comments (*since=NotSet*)

Calls `GET /repos/:owner/:repo/pulls/:number/comments`

Parameters **since** – datetime.datetime format YYYY-MM-DDTHH:MM:SSZ

Return type `github.PaginatedList.PaginatedList` of `github.PullRequestComment.PullRequestComment`

get_single_review_comments (*id*)

Calls `GET /repos/:owner/:repo/pulls/:number/review/:id/comments`

Parameters **id** – integer

Return type `github.PaginatedList.PaginatedList` of `github.PullRequestComment.PullRequestComment`

get_commits ()

Calls `GET /repos/:owner/:repo/pulls/:number/commits`

Return type `github.PaginatedList.PaginatedList` of `github.Commit.Commit`

get_files ()

Calls `GET /repos/:owner/:repo/pulls/:number/files`

Return type `github.PaginatedList.PaginatedList` of `github.File.File`

get_issue_comment (*id*)

Calls GET /repos/:owner/:repo/issues/comments/:id

Parameters **id** – integer

Return type *github.IssueComment.IssueComment*

get_issue_comments ()

Calls GET /repos/:owner/:repo/issues/:number/comments

Return type *github.PaginatedList.PaginatedList* of *github.IssueComment.IssueComment*

get_review (*id*)

Calls GET /repos/:owner/:repo/pulls/:number/reviews/:id

Parameters **id** – integer

Return type *github.PullRequestReview.PullRequestReview*

get_reviews ()

Calls GET /repos/:owner/:repo/pulls/:number/reviews

Return type *github.PaginatedList.PaginatedList* of *github.PullRequestReview.PullRequestReview*

get_review_requests ()

Calls GET /repos/:owner/:repo/pulls/:number/requested_reviewers

Return type tuple of *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser* and of *github.PaginatedList.PaginatedList* of *github.Team.Team*

get_labels ()

Calls GET /repos/:owner/:repo/issues/:number/labels

Return type *github.PaginatedList.PaginatedList* of *github.Label.Label*

add_to_labels (**labels*)

Calls POST /repos/:owner/:repo/issues/:number/labels

Parameters **label** – *github.Label.Label* or string

Return type None

delete_labels ()

Calls DELETE /repos/:owner/:repo/issues/:number/labels

Return type None

remove_from_labels (*label*)

Calls DELETE /repos/:owner/:repo/issues/:number/labels/:name

Parameters **label** – *github.Label.Label* or string

Return type None

set_labels (**labels*)

Calls PUT /repos/:owner/:repo/issues/:number/labels

Parameters `labels` – list of `github.Label.Label` or strings

Return type None

`is_merged()`

Calls GET /repos/:owner/:repo/pulls/:number/merge

Return type bool

`merge(commit_message=NotSet, commit_title=NotSet, merge_method=NotSet, sha=NotSet)`

Calls PUT /repos/:owner/:repo/pulls/:number/merge

Parameters `commit_message` – string

Return type `github.PullRequestMergeStatus.PullRequestMergeStatus`

2.4.50 PullRequestComment

class `github.PullRequestComment.PullRequestComment`

This class represents PullRequestComments. The reference can be found here <http://developer.github.com/v3/pulls/comments/>

body

Type string

commit_id

Type string

created_at

Type datetime.datetime

diff_hunk

Type string

id

Type integer

in_reply_to_id

Type integer

original_commit_id

Type string

original_position

Type integer

path

Type string

position

Type integer

pull_request_url

Type string

updated_at

Type `datetime.datetime`
url
Type `string`
html_url
Type `string`
user
Type `github.NamedUser.NamedUser`
delete()
Calls `DELETE /repos/:owner/:repo/pulls/comments/:number`
Return type `None`
edit(*body*)
Calls `PATCH /repos/:owner/:repo/pulls/comments/:number`
Parameters **body** – `string`
Return type `None`
get_reactions()
Calls `GET /repos/:owner/:repo/pulls/comments/:number/reactions`
Returns
class `github.PaginatedList.PaginatedList` of `github.Reaction.Reaction`.
create_reaction(*reaction_type*)
Calls `POST /repos/:owner/:repo/pulls/comments/:number/reactions`
Parameters **reaction_type** – `string`
Return type `github.Reaction.Reaction`

2.4.51 PullRequestMergeStatus

class `github.PullRequestMergeStatus.PullRequestMergeStatus`

This class represents PullRequestMergeStatuses. The reference can be found here <http://developer.github.com/v3/pulls/#get-if-a-pull-request-has-been-merged>

merged

Type `bool`

message

Type `string`

sha

Type `string`

2.4.52 PullRequestPart

class `github.PullRequestPart.PullRequestPart`

This class represents PullRequestParts

label

Type string

ref

Type string

repo

Type `github.Repository.Repository`

sha

Type string

user

Type `github.NamedUser.NamedUser`

2.4.53 PullRequestReview

class `github.PullRequestReview.PullRequestReview`

This class represents PullRequestReviews. The reference can be found here <https://developer.github.com/v3/pulls/reviews/>

id

Type integer

user

Type `github.NamedUser.NamedUser`

body

Type string

commit_id

Type string

state

Type string

url

Type string

html_url

Type string

pull_request_url

Type string

submitted_at

Type `datetime.datetime`

2.4.54 Rate

class `github.Rate.Rate`

This class represents Rates. The reference can be found here http://developer.github.com/v3/rate_limit

limit

Type integer

remaining

Type integer

reset

Type `datetime.datetime`

2.4.55 RateLimit

class `github.RateLimit.RateLimit`

This class represents RateLimits. The reference can be found here http://developer.github.com/v3/rate_limit

rate

Type `class:github.Rate.Rate`

2.4.56 Reaction

class `github.Reaction.Reaction`

This class represents Reactions. The reference can be found here <https://developer.github.com/v3/reactions/>

content

Type string

created_at

Type `datetime.datetime`

id

Type integer

user

Type `github.NamedUser.NamedUser`

delete()

Calls `DELETE /reactions/:id`

Return type None

2.4.57 Repository

class `github.Repository.Repository`

This class represents Repositories. The reference can be found here <http://developer.github.com/v3/repos/>

allow_merge_commit

Type bool

allow_rebase_merge

Type bool

allow_squash_merge

Type bool

archived

Type bool

archive_url

Type string

assignees_url

Type string

blobs_url

Type string

branches_url

Type string

clone_url

Type string

collaborators_url

Type string

comments_url

Type string

commits_url

Type string

compare_url

Type string

contents_url

Type string

contributors_url

Type string

created_at

Type datetime.datetime

default_branch

Type string

description

Type string

downloads_url

Type string

events_url
 Type string

fork
 Type bool

forks
 Type integer

forks_count
 Type integer

forks_url
 Type string

full_name
 Type string

git_commits_url
 Type string

git_refs_url
 Type string

git_tags_url
 Type string

git_url
 Type string

has_downloads
 Type bool

has_issues
 Type bool

has_projects
 Type bool

has_wiki
 Type bool

homepage
 Type string

hooks_url
 Type string

html_url
 Type string

id
 Type integer

issue_comment_url
Type string

issue_events_url
Type string

issues_url
Type string

keys_url
Type string

labels_url
Type string

language
Type string

languages_url
Type string

master_branch
Type string

merges_url
Type string

milestones_url
Type string

mirror_url
Type string

name
Type string

network_count
Type integer

notifications_url
Type string

open_issues
Type integer

open_issues_count
Type integer

organization
Type *github.Organization.Organization*

owner
Type *github.NamedUser.NamedUser*

parent
Type *github.Repository.Repository*

permissions
Type *github.Permissions.Permissions*

private
Type bool

pulls_url
Type string

pushed_at
Type datetime.datetime

size
Type integer

source
Type *github.Repository.Repository*

ssh_url
Type string

stargazers_count
Type integer

stargazers_url
Type string

statuses_url
Type string

subscribers_url
Type string

subscribers_count
Type integer

subscription_url
Type string

svn_url
Type string

tags_url
Type string

teams_url
Type string

topics
Type list of strings

trees_url

Type string

updated_at

Type datetime.datetime

url

Type string

watchers

Type integer

watchers_count

Type integer

add_to_collaborators (*collaborator, permission=NotSet*)

Calls `PUT /repos/:owner/:repo/collaborators/:user`

Parameters

- **collaborator** – string or `github.NamedUser.NamedUser`
- **permission** – string ‘pull’, ‘push’ or ‘admin’

Return type None

compare (*base, head*)

Calls `GET /repos/:owner/:repo/compare/:base...:head`

Parameters

- **base** – string
- **head** – string

Return type `github.Comparison.Comparison`

create_git_blob (*content, encoding*)

Calls `POST /repos/:owner/:repo/git/blobs`

Parameters

- **content** – string
- **encoding** – string

Return type `github.GitBlob.GitBlob`

create_git_commit (*message, tree, parents, author=NotSet, committer=NotSet*)

Calls `POST /repos/:owner/:repo/git/commits`

Parameters

- **message** – string
- **tree** – `github.GitTree.GitTree`
- **parents** – list of `github.GitCommit.GitCommit`
- **author** – `github.InputGitAuthor.InputGitAuthor`
- **committer** – `github.InputGitAuthor.InputGitAuthor`

Return type *github.GitCommit.GitCommit*

create_git_ref (*ref, sha*)

Calls POST /repos/:owner/:repo/git/refs

Parameters

- **ref** – string
- **sha** – string

Return type *github.GitRef.GitRef*

create_git_release (*tag, name, message, draft=False, prerelease=False, target_commitish=NotSet*)

Calls POST /repos/:owner/:repo/releases

Parameters

- **tag** – string
- **name** – string
- **message** – string
- **draft** – bool
- **prerelease** – bool
- **target_commitish** – string or *github.Branch.Branch* or *github.Commit.Commit* or *github.GitCommit.GitCommit*

Return type *github.GitRelease.GitRelease*

create_git_tag (*tag, message, object, type, tagger=NotSet*)

Calls POST /repos/:owner/:repo/git/tags

Parameters

- **tag** – string
- **message** – string
- **object** – string
- **type** – string
- **tagger** – *github.InputGitAuthor.InputGitAuthor*

Return type *github.GitTag.GitTag*

create_git_tree (*tree, base_tree=NotSet*)

Calls POST /repos/:owner/:repo/git/trees

Parameters

- **tree** – list of *github.InputGitTreeElement.InputGitTreeElement*
- **base_tree** – *github.GitTree.GitTree*

Return type *github.GitTree.GitTree*

create_hook (*name, config, events=NotSet, active=NotSet*)

Calls POST /repos/:owner/:repo/hooks

Parameters

- **name** – string
- **config** – dict
- **events** – list of string
- **active** – bool

Return type *github.Hook.Hook*

create_issue (*title*, *body=NotSet*, *assignee=NotSet*, *milestone=NotSet*, *labels=NotSet*, *assignees=NotSet*)

Calls POST /repos/:owner/:repo/issues

Parameters

- **title** – string
- **body** – string
- **assignee** – string or *github.NamedUser.NamedUser*
- **assignees** – list (of string or *github.NamedUser.NamedUser*)
- **milestone** – *github.Milestone.Milestone*
- **labels** – list of *github.Label.Label*

Return type *github.Issue.Issue*

create_key (*title*, *key*, *read_only=False*)

Calls POST /repos/:owner/:repo/keys

Parameters

- **title** – string
- **key** – string
- **read_only** – bool

Return type *github.RepositoryKey.RepositoryKey*

create_label (*name*, *color*, *description=NotSet*)

Calls POST /repos/:owner/:repo/labels

Parameters

- **name** – string
- **color** – string
- **description** – string

Return type *github.Label.Label*

create_milestone (*title*, *state=NotSet*, *description=NotSet*, *due_on=NotSet*)

Calls POST /repos/:owner/:repo/milestones

Parameters

- **title** – string
- **state** – string
- **description** – string
- **due_on** – datetime

Return type *github.Milestone.Milestone*

create_pull (*args, **kws)

Calls POST /repos/:owner/:repo/pulls

Parameters

- **title** – string
- **body** – string
- **issue** – *github.Issue.Issue*
- **base** – string
- **head** – string
- **maintainer_can_modify** – bool

Return type *github.PullRequest.PullRequest*

create_source_import (vcs, vcs_url, vcs_username=NotSet, vcs_password=NotSet)

Calls PUT /repos/:owner/:repo/import

Parameters

- **vcs** – string
- **vcs_url** – string
- **vcs_username** – string
- **vcs_password** – string

Return type *github.SourceImport.SourceImport*

delete ()

Calls DELETE /repos/:owner/:repo

Return type None

edit (name=None, description=NotSet, homepage=NotSet, private=NotSet, has_issues=NotSet, has_projects=NotSet, has_wiki=NotSet, has_downloads=NotSet, default_branch=NotSet, allow_squash_merge=NotSet, allow_merge_commit=NotSet, allow_rebase_merge=NotSet, archived=NotSet)

Calls PATCH /repos/:owner/:repo

Parameters

- **name** – string
- **description** – string
- **homepage** – string
- **private** – bool
- **has_issues** – bool
- **has_projects** – bool
- **has_wiki** – bool
- **has_downloads** – bool
- **default_branch** – string

- `allow_squash_merge` – bool
- `allow_merge_commit` – bool
- `allow_rebase_merge` – bool
- `archived` – bool. Unarchiving repositories is currently not supported through API (<https://developer.github.com/v3/repos/#edit>)

Return type None

`get_archive_link` (*archive_format*, *ref=NotSet*)

Calls GET /repos/:owner/:repo/:archive_format/:ref

Parameters

- `archive_format` – string
- `ref` – string

Return type string

`get_assignees` ()

Calls GET /repos/:owner/:repo/assignees

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

`get_branch` (*branch*)

Calls GET /repos/:owner/:repo/branches/:branch

Parameters `branch` – string

Return type *github.Branch.Branch*

`get_protected_branch` (*branch*)

Calls GET /repos/:owner/:repo/branches/:branch

Parameters `branch` – string

Return type *github.Branch.Branch*

`get_branches` ()

Calls GET /repos/:owner/:repo/branches

Return type *github.PaginatedList.PaginatedList* of *github.Branch.Branch*

`get_collaborators` ()

Calls GET /repos/:owner/:repo/collaborators

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

`get_comment` (*id*)

Calls GET /repos/:owner/:repo/comments/:id

Parameters `id` – integer

Return type *github.CommitComment.CommitComment*

`get_comments` ()

Calls `GET /repos/:owner/:repo/comments`

Return type `github.PaginatedList.PaginatedList` of `github.CommitComment.CommitComment`

get_commit (*sha*)

Calls `GET /repos/:owner/:repo/commits/:sha`

Parameters `sha` – string

Return type `github.Commit.Commit`

get_commits (*sha=NotSet, path=NotSet, since=NotSet, until=NotSet, author=NotSet*)

Calls `GET /repos/:owner/:repo/commits`

Parameters

- `sha` – string
- `path` – string
- `since` – `datetime.datetime`
- `until` – `datetime.datetime`
- `author` – string or `github.NamedUser.NamedUser` or `github.AuthenticatedUser.AuthenticatedUser`

Return type `github.PaginatedList.PaginatedList` of `github.Commit.Commit`

get_contents (*path, ref=NotSet*)

Calls `GET /repos/:owner/:repo/contents/:path`

Parameters

- `path` – string
- `ref` – string

Return type `github.ContentFile.ContentFile`

get_file_contents (*path, ref=NotSet*)

Calls `GET /repos/:owner/:repo/contents/:path`

Parameters

- `path` – string
- `ref` – string

Return type `github.ContentFile.ContentFile`

create_file (*path, message, content, branch=NotSet, committer=NotSet, author=NotSet*)

Create a file in this repository.

Calls `PUT /repos/:owner/:repo/contents/:path`

Parameters

- `path` – string, (required), path of the file in the repository
- `message` – string, (required), commit message
- `content` – string, (required), the actual data in the file

- **branch** – string, (optional), branch to create the commit on. Defaults to the default branch of the repository
- **committer** – InputGitAuthor, (optional), if no information is given the authenticated user’s information will be used. You must specify both a name and email.
- **author** – InputGitAuthor, (optional), if omitted this will be filled in with committer information. If passed, you must specify both a name and email.

Return type

```
{ 'content': ContentFile;, 'commit': Commit }
```

update_file (*path, message, content, sha, branch=NotSet, committer=NotSet, author=NotSet*)

This method updates a file in a repository

Calls [PUT /repos/:owner/:repo/contents/:path](#)

Parameters

- **path** – string, Required. The content path.
- **message** – string, Required. The commit message.
- **content** – string, Required. The updated file content, Base64 encoded.
- **sha** – string, Required. The blob SHA of the file being replaced.
- **branch** – string. The branch name. Default: the repository’s default branch (usually master)
- **committer** – InputGitAuthor, (optional), if no information is given the authenticated user’s information will be used. You must specify both a name and email.
- **author** – InputGitAuthor, (optional), if omitted this will be filled in with committer information. If passed, you must specify both a name and email.

Return type

```
{ 'content': ContentFile;, 'commit': Commit }
```

delete_file (*path, message, sha, branch=NotSet, committer=NotSet, author=NotSet*)

This method deletes a file in a repository

Calls [DELETE /repos/:owner/:repo/contents/:path](#)

Parameters

- **path** – string, Required. The content path.
- **message** – string, Required. The commit message.
- **sha** – string, Required. The blob SHA of the file being replaced.
- **branch** – string. The branch name. Default: the repository’s default branch (usually master)
- **committer** – InputGitAuthor, (optional), if no information is given the authenticated user’s information will be used. You must specify both a name and email.
- **author** – InputGitAuthor, (optional), if omitted this will be filled in with committer information. If passed, you must specify both a name and email.

Return type

```
{ 'content': null;, 'commit': Commit }
```

get_dir_contents (*path, ref=NotSet*)

Calls GET /repos/:owner/:repo/contents/:path

Parameters

- **path** – string
- **ref** – string

Return type list of *github.ContentFile.ContentFile*

get_contributors ()

Calls GET /repos/:owner/:repo/contributors

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

get_download (*id*)

Calls GET /repos/:owner/:repo/downloads/:id

Parameters **id** – integer

Return type *github.Download.Download*

get_downloads ()

Calls GET /repos/:owner/:repo/downloads

Return type *github.PaginatedList.PaginatedList* of *github.Download.Download*

get_events ()

Calls GET /repos/:owner/:repo/events

Return type *github.PaginatedList.PaginatedList* of *github.Event.Event*

get_forks ()

Calls GET /repos/:owner/:repo/forks

Return type *github.PaginatedList.PaginatedList* of *github.Repository.Repository*

get_git_blob (*sha*)

Calls GET /repos/:owner/:repo/git/blobs/:sha

Parameters **sha** – string

Return type *github.GitBlob.GitBlob*

get_git_commit (*sha*)

Calls GET /repos/:owner/:repo/git/commits/:sha

Parameters **sha** – string

Return type *github.GitCommit.GitCommit*

get_git_ref (*ref*)

Calls GET /repos/:owner/:repo/git/refs/:ref

Parameters **ref** – string

Return type *github.GitRef.GitRef*

get_git_refs ()

Calls `GET /repos/:owner/:repo/git/refs`

Return type `github.PaginatedList.PaginatedList` of `github.GitRef.GitRef`

get_git_tag (*sha*)

Calls `GET /repos/:owner/:repo/git/tags/:sha`

Parameters **sha** – string

Return type `github.GitTag.GitTag`

get_git_tree (*sha, recursive=NotSet*)

Calls `GET /repos/:owner/:repo/git/trees/:sha`

Parameters

- **sha** – string
- **recursive** – bool

Return type `github.GitTree.GitTree`

get_hook (*id*)

Calls `GET /repos/:owner/:repo/hooks/:id`

Parameters **id** – integer

Return type `github.Hook.Hook`

get_hooks ()

Calls `GET /repos/:owner/:repo/hooks`

Return type `github.PaginatedList.PaginatedList` of `github.Hook.Hook`

get_issue (*number*)

Calls `GET /repos/:owner/:repo/issues/:number`

Parameters **number** – integer

Return type `github.Issue.Issue`

get_issues (*milestone=NotSet, state=NotSet, assignee=NotSet, mentioned=NotSet, labels=NotSet, sort=NotSet, direction=NotSet, since=NotSet, creator=NotSet*)

Calls `GET /repos/:owner/:repo/issues`

Parameters

- **milestone** – `github.Milestone.Milestone` or “none” or “*”
- **state** – string
- **assignee** – string or `github.NamedUser.NamedUser` or “none” or “*”
- **mentioned** – `github.NamedUser.NamedUser`
- **labels** – list of `github.Label.Label`
- **sort** – string
- **direction** – string
- **since** – `datetime.datetime`
- **creator** – string or `github.NamedUser.NamedUser`

Return type *github.PaginatedList.PaginatedList* of *github.Issue.Issue*

get_issues_comments (*sort=NotSet, direction=NotSet, since=NotSet*)

Calls GET /repos/:owner/:repo/issues/comments

Parameters

- **sort** – string
- **direction** – string
- **since** – datetime.datetime

Return type *github.PaginatedList.PaginatedList* of *github.IssueComment.IssueComment*

get_issues_event (*id*)

Calls GET /repos/:owner/:repo/issues/events/:id

Parameters **id** – integer

Return type *github.IssueEvent.IssueEvent*

get_issues_events ()

Calls GET /repos/:owner/:repo/issues/events

Return type *github.PaginatedList.PaginatedList* of *github.IssueEvent.IssueEvent*

get_key (*id*)

Calls GET /repos/:owner/:repo/keys/:id

Parameters **id** – integer

Return type *github.RepositoryKey.RepositoryKey*

get_keys ()

Calls GET /repos/:owner/:repo/keys

Return type *github.PaginatedList.PaginatedList* of *github.RepositoryKey.RepositoryKey*

get_label (*name*)

Calls GET /repos/:owner/:repo/labels/:name

Parameters **name** – string

Return type *github.Label.Label*

get_labels ()

Calls GET /repos/:owner/:repo/labels

Return type *github.PaginatedList.PaginatedList* of *github.Label.Label*

get_languages ()

Calls GET /repos/:owner/:repo/languages

Return type dict of string to integer

get_license ()

Calls GET /repos/:owner/:repo/license

Return type *github.ContentFile.ContentFile*

get_milestone (*number*)

Calls GET /repos/:owner/:repo/milestones/:number

Parameters **number** – integer

Return type *github.Milestone.Milestone*

get_milestones (*state=NotSet, sort=NotSet, direction=NotSet*)

Calls GET /repos/:owner/:repo/milestones

Parameters

- **state** – string
- **sort** – string
- **direction** – string

Return type *github.PaginatedList.PaginatedList* of *github.Milestone.Milestone*

get_network_events ()

Calls GET /networks/:owner/:repo/events

Return type *github.PaginatedList.PaginatedList* of *github.Event.Event*

get_pull (*number*)

Calls GET /repos/:owner/:repo/pulls/:number

Parameters **number** – integer

Return type *github.PullRequest.PullRequest*

get_pulls (*state=NotSet, sort=NotSet, direction=NotSet, base=NotSet, head=NotSet*)

Calls GET /repos/:owner/:repo/pulls

Parameters

- **state** – string
- **sort** – string
- **direction** – string
- **base** – string
- **head** – string

Return type *github.PaginatedList.PaginatedList* of *github.PullRequest.PullRequest*

get_pulls_comments (*sort=NotSet, direction=NotSet, since=NotSet*)

Calls GET /repos/:owner/:repo/pulls/comments

Parameters

- **sort** – string
- **direction** – string
- **since** – datetime.datetime

Return type *github.PaginatedList.PaginatedList* of *github.PullRequestComment.PullRequestComment*

get_pulls_review_comments (*sort=NotSet, direction=NotSet, since=NotSet*)

Calls GET /repos/:owner/:repo/pulls/comments

Parameters

- **sort** – string
- **direction** – string
- **since** – datetime.datetime

Return type *github.PaginatedList.PaginatedList* of *github.PullRequestComment.PullRequestComment*

get_readme (*ref=NotSet*)

Calls GET /repos/:owner/:repo/readme

Parameters **ref** – string

Return type *github.ContentFile.ContentFile*

get_source_import ()

Calls ‘GET /repos/:owner/:repo/import https://developer.github.com/v3/migration/source_imports/#get-import-progress‘_

Return type *github.SourceImport.SourceImport*

get_stargazers ()

Calls GET /repos/:owner/:repo/stargazers

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

get_stargazers_with_dates ()

Calls GET /repos/:owner/:repo/stargazers

Return type *github.PaginatedList.PaginatedList* of *github.Stargazer.Stargazer*

get_stats_contributors ()

Calls GET /repos/:owner/:repo/stats/contributors

Return type None or list of *github.StatsContributor.StatsContributor*

get_stats_commit_activity ()

Calls GET /repos/:owner/:repo/stats/commit_activity

Return type None or list of *github.StatsCommitActivity.StatsCommitActivity*

get_stats_code_frequency ()

Calls GET /repos/:owner/:repo/stats/code_frequency

Return type None or list of *github.StatsCodeFrequency.StatsCodeFrequency*

get_stats_participation ()

Calls GET /repos/:owner/:repo/stats/participation

Return type None or *github.StatsParticipation.StatsParticipation*

get_stats_punch_card()

Calls GET /repos/:owner/:repo/stats/punch_card

Return type None or *github.StatsPunchCard.StatsPunchCard*

get_subscribers()

Calls GET /repos/:owner/:repo/subscribers

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

get_tags()

Calls GET /repos/:owner/:repo/tags

Return type *github.PaginatedList.PaginatedList* of *github.Tag.Tag*

get_releases()

Calls GET /repos/:owner/:repo/releases

Return type *github.PaginatedList.PaginatedList* of *github.GitRelease.GitRelease*

get_release(id)

Calls GET /repos/:owner/:repo/releases/:id

Parameters **id** – int (release id), str (tag name)

Return type None or *github.GitRelease.GitRelease*

get_latest_release()

Calls ‘GET /repos/:owner/:repo/releases/latest <https://developer.github.com/v3/repos/releases/#get-the-latest-release>

Return type *github.GitRelease.GitRelease*

get_teams()

Calls GET /repos/:owner/:repo/teams

Return type *github.PaginatedList.PaginatedList* of *github.Team.Team*

get_topics()

Calls GET /repos/:owner/:repo/topics

Return type list of strings

get_watchers()

Calls GET /repos/:owner/:repo/watchers

Return type *github.PaginatedList.PaginatedList* of *github.NamedUser.NamedUser*

has_in_assignees(assignee)

Calls GET /repos/:owner/:repo/assignees/:assignee

Parameters **assignee** – string or *github.NamedUser.NamedUser*

Return type bool

has_in_collaborators (*collaborator*)

Calls `GET /repos/:owner/:repo/collaborators/:user`

Parameters **collaborator** – string or `github.NamedUser.NamedUser`

Return type `bool`

legacy_search_issues (*state, keyword*)

Calls `GET /legacy/issues/search/:owner/:repository/:state/:keyword`

Parameters

- **state** – “open” or “closed”
- **keyword** – string

Return type `github.PaginatedList.PaginatedList` of `github.Issue.Issue`

merge (*base, head, commit_message=NotSet*)

Calls `POST /repos/:owner/:repo/merges`

Parameters

- **base** – string
- **head** – string
- **commit_message** – string

Return type `github.Commit.Commit`

protect_branch (*branch, enabled, enforcement_level=NotSet, contexts=NotSet*)

Calls `PATCH /repos/:owner/:repo/branches/:branch`

Parameters

- **branch** – string
- **enabled** – boolean
- **enforcement_level** – string
- **contexts** – list of strings

Return type `None`

replace_topics (*topics*)

Calls `PUT /repos/:owner/:repo/topics`

Parameters **topics** – list of strings

Return type `None`

remove_from_collaborators (*collaborator*)

Calls `DELETE /repos/:owner/:repo/collaborators/:user`

Parameters **collaborator** – string or `github.NamedUser.NamedUser`

Return type `None`

subscribe_to_hub (*event, callback, secret=NotSet*)

Calls `POST /hub`

Parameters

- **event** – string
- **callback** – string
- **secret** – string

Return type None

unsubscribe_from_hub (*event, callback*)

Calls POST /hub

Parameters

- **event** – string
- **callback** – string
- **secret** – string

Return type None

2.4.58 RepositoryKey

class github.RepositoryKey.**RepositoryKey**

This class represents RepositoryKeys. The reference can be found here <http://developer.github.com/v3/repos/keys/>

created_at

Type datetime.datetime

id

Type integer

key

Type string

title

Type string

url

Type string

verified

Type bool

read_only

Type bool

delete ()

Calls DELETE /repos/:owner/:repo/keys/:id

Return type None

2.4.59 SourceImport

class github.SourceImport.SourceImport

This class represents SourceImports. The reference can be found here https://developer.github.com/v3/migration/source_imports/

authors_count

Type integer

authors_url

Type string

has_large_files

Type bool

html_url

Type string

large_files_count

Type integer

large_files_size

Type integer

repository_url

Type string

status

Type string

status_text

Type string

url

Type string

use_lfs

Type string

vcs

Type string

vcs_url

Type string

2.4.60 Stargazer

class github.Stargazer.Stargazer

This class represents Stargazers. The reference can be found here <https://developer.github.com/v3/activity/starring/#alternative-response-with-star-creation-timestamps>

starred_at

Type datetime.datetime

user

Type github.NamedUser

2.4.61 StatsCodeFrequency

class github.StatsCodeFrequency.**StatsCodeFrequency**

This class represents statistics of StatsCodeFrequencies. The reference can be found here <http://developer.github.com/v3/repos/statistics/#get-the-number-of-additions-and-deletions-per-week>

week

Type datetime.datetime

additions

Type int

deletions

Type int

2.4.62 StatsCommitActivity

class github.StatsCommitActivity.**StatsCommitActivity**

This class represents StatsCommitActivities. The reference can be found here <http://developer.github.com/v3/repos/statistics/#get-the-last-year-of-commit-activity-data>

week

Type datetime.datetime

total

Type int

days

Type list of int

2.4.63 StatsContributor

class github.StatsContributor.**StatsContributor**

This class represents StatsContributors. The reference can be found here <http://developer.github.com/v3/repos/statistics/#get-contributors-list-with-additions-deletions-and-commit-counts>

class **Week** (*requester, headers, attributes, completed*)

This class represents weekly statistics of a contributor.

w

Type datetime.datetime

a

Type int

d

Type int

c

Type int

authorType *github.NamedUser.NamedUser***total**

Type int

weeksType list of *Week*

2.4.64 StatsParticipation

class `github.StatsParticipation.StatsParticipation`

This class represents StatsParticipations. The reference can be found here <http://developer.github.com/v3/repos/statistics/#get-the-weekly-commit-count-for-the-repo-owner-and-everyone-else>

all

Type list of int

owner

Type list of int

2.4.65 StatsPunchCard

class `github.StatsPunchCard.StatsPunchCard`

This class represents StatsPunchCards. The reference can be found here <http://developer.github.com/v3/repos/statistics/#get-the-number-of-commits-per-hour-in-each-day>

get (*day, hour*)

Get a specific element :param day: int :param hour: int :rtype: int

2.4.66 Status

class `github.Status.Status`

This class represents Statuses. The reference can be found here <https://status.github.com/api>

status

Type string

last_updatedType `datetime.datetime`

2.4.67 StatusMessage

class `github.StatusMessage.StatusMessage`

This class represents StatusMessages. The reference can be found here <https://status.github.com/api>

body

Type string

status

Type string
created_on
Type datetime.datetime

2.4.68 Tag

class github.Tag.**Tag**

This class represents Tags. The reference can be found here <https://developer.github.com/v3/repos/#list-tags>

commit
Type *github.Commit.Commit*
name
Type string
tarball_url
Type string
zipball_url
Type string

2.4.69 Team

class github.Team.**Team**

This class represents Teams. The reference can be found here <http://developer.github.com/v3/orgs/teams/>

id
Type integer
members_count
Type integer
members_url
Type string
name
Type string
description
Type string
permission
Type string
repos_count
Type integer
repositories_url
Type string
slug

Type string

url

Type string

organization

Type *github.Organization.Organization*

privacy

Type string

add_to_members (*member*)

This API call is deprecated. Use *add_membership* instead. <https://developer.github.com/v3/teams/members/#deprecation-notice-1>

Calls PUT /teams/:id/members/user

Parameters **member** – *github.NamedUser.NamedUser*

Return type None

add_membership (*member*, *role=NotSet*)

Calls PUT /teams/:id/memberships/user

Parameters

- **member** – *github.Nameduser.NamedUser*
- **role** – string

Return type None

add_to_repos (*repo*)

Calls PUT /teams/:id/repos/:org:repo

Parameters **repo** – *github.Repository.Repository*

Return type None

set_repo_permission (*repo*, *permission*)

Calls PUT /teams/:id/repos/:org:repo

Parameters

- **repo** – *github.Repository.Repository*
- **permission** – string

Return type None

delete ()

Calls DELETE /teams/:id

Return type None

edit (*name*, *description=NotSet*, *permission=NotSet*, *privacy=NotSet*)

Calls PATCH /teams/:id

Parameters

- **name** – string
- **description** – string

- **permission** – string
- **privacy** – string

Return type None

get_members (*role=NotSet*)

Calls `GET /teams/:id/members`

Parameters **role** – string

Return type `github.PaginatedList.PaginatedList` of `github.NamedUser.NamedUser`

get_repos ()

Calls `GET /teams/:id/repos`

Return type `github.PaginatedList.PaginatedList` of `github.Repository.Repository`

has_in_members (*member*)

Calls `GET /teams/:id/members/:user`

Parameters **member** – `github.NamedUser.NamedUser`

Return type bool

has_in_repos (*repo*)

Calls `GET /teams/:id/repos/:owner/:repo`

Parameters **repo** – `github.Repository.Repository`

Return type bool

remove_membership (*member*)

Calls `DELETE /teams/:team_id/memberships/:username <https://developer.github.com/v3/teams/members/#remove-team-membership>`

Parameters **member** –

Returns

remove_from_members (*member*)

This API call is deprecated. Use `remove_membership` instead: <https://developer.github.com/v3/teams/members/#deprecation-notice-2>

Calls `DELETE /teams/:id/members/:user`

Parameters **member** – `github.NamedUser.NamedUser`

Return type None

remove_from_repos (*repo*)

Calls `DELETE /teams/:id/repos/:owner/:repo`

Parameters **repo** – `github.Repository.Repository`

Return type None

2.4.70 UserKey

class `github.UserKey.UserKey`

This class represents UserKeys. The reference can be found here <http://developer.github.com/v3/users/keys/>

id

Type integer

key

Type string

title

Type string

url

Type string

verified

Type bool

delete()

Calls `DELETE /user/keys/:id`

Return type None

3.1 Stable versions

3.1.1 Version 1.40 (June 26, 2018)

- Major enhancement: use requests for HTTP instead of httplib (#664) (9aed19dd)
- Test Framework improvement (#795) (faa8f205)
- Handle HTTP 202 HEAD & GET with a retry (#791) (3aead158)
- Fix github API requests after asset upload (#771) (8bdac23c)
- Add remove_membership() method to Teams class (#807) (817f2230)
- Add check-in to projects using PyGithub (#814) (05f49a59)
- Include target_commitish in GitRelease (#788) (ba5bf2d7)
- Fix asset upload timeout, increase default timeout from 10s to 15s (#793) (140c6480)
- Fix Team.description (#797) (0e8ae376)
- Fix Content-Length invalid headers exception (#787) (23395f5f)
- Remove NamedUser.contributions (#774) (a519e467)
- Add ability to skip SSL cert verification for Github Enterprise (#758) (85a9124b)
- Correct Repository.get_git_tree recursive use (#767) (bd0cf309)
- Re-work PullRequest reviewer request (#765) (e2e29918)
- Add support for team privacy (#763) (1f23c06a)
- Add support for organization outside collaborators (#533) (c4446996)
- PullRequest labels should use Issues URL (#754) (678b6b20)
- Support labels for PullRequests (#752) (a308dc92)

- Add `get_organizations()` (#748) (1e0150b5)

3.1.2 Version 1.39 (April 10, 2018)

- Add documentation to `github.Repository.Repository.create_git_release()` (#747) (a769c2ff)
- Add `add_to_members()` and `remove_from_membership()` (#741) (4da483d1)
- Documentation: clarify semantics of `get_comments` (#743) (fec3c943)
- Add `download_url` to `ContentFile`, closes #575 (ca6fbc45)
- Add `PullRequestComment.in_reply_to_id` (#718) (eaa6a508)
- Add team privacy parameter to create team (#702) (5cb5ab71)
- Implement License API (#734) (b54ccc78)
- Fix delete method for `RepositoryKey` (911bf615)
- Remove edit for `UserKey` (722f2534)
- Labels API: support description (#738) (42e75938)
- Added `Issue.as_pull_request()` and `PullRequest.as_issue()` (#630) (6bf2acc7)
- Documentation: sort the Github Objects (#735) (1497e826)
- Add support for getting PR single review's comments. (#670) (612c3500)
- Update the `RepositoryKey` class (#530) (5e8c6832)
- Added `since` to PR review comments get (#577) (d8508285)
- Remove some duplicate attributes introduced in #522 (566b28d3)
- Added `tarball_url`, `zipball_url`, `prerelease` and `draft` property (#522) (c76e67b7)
- Source Import API (#673) (864c663a)

3.1.3 Version 1.38 (March 21, 2018)

- Updated `readthedocs`, `PyPI` to reflect latest version
- Added option to create review for Pull request (#662) (162f0397)
- Depreciate legacy search API (3cd176e3)
- Filter team members by role (#491) (10ee17a2)
- Add `url` attribute to `PullRequestReview` object (#731) (0fb176fd)
- Added `target_commitish` option to `Repository.create_git_release()` (#625) (0f0a7d4e)
- Fix broken Github reference link in class docstrings (a32a17bf)
- Add hook support for organizations (#729) (c7f6563c)
- Get organization from the team (#590) (d9c5a07f)
- Added `search_commits` (#727) (aa556f85)
- Collaborator site admin (#719) (f8b23505)
- Fix `add_to_watched` for `AuthenticatedUser` (#716) (6109eb3c)

3.1.4 Version 1.37 (March 03, 2018)

- Add `__eq__` and `__hash__` to `NamedUser` (#706) (8a13b274)
- Add maintainer can modify flag to create pull request (#703) (0e5a1d1d)
- Fix typo in `Design.md` (#701) (98d32af4)
- Add role parameter to `Team.add_membership` method (#638) (01ab4cc6)
- Add `add_membership` testcase (#637) (5a1424bb)

3.1.5 Version 1.36 (February 02, 2018)

- Fix changelog generation (5d911e22)
- Add collaborator permission support (#699) (167f85ef)
- Use `datetime` object in `create_milestone` (#698) (cef98416)
- Fix date format for milestone creation (#593) (e671fdd0)
- Remove the default “null” input send during GET request (#691) (cbfe8d0f)
- Updated `PullRequest` reviewer request according to API changes (#690) (5c9c2f75)
- make `created_at/published_at` attrs available for `Release` objects (#689) (2f9b1e01)
- Add `committer/author` to `Repository.delete_file` (#678) (3baa682c)
- Add method to get latest release of a repository (#609) (45d18436)
- Add `permissions` field to `NamedUser` (#676) (6cfe46b7)
- Fix all pep8 coding conventions (6bc804dc)
- Add new params for `/users/:user/repos` endpoint (89834a9b)
- Add support for changing PR head commit (#632) (3f77e537)
- Use `print()` syntax in `README` (#681) (c5988c39)
- Add PyPI badge and installation instructions to `README` (#682) (3726f686)
- Drop support for EOL Python 2.5-2.6 and 3.2-3.3 (#674) (6735be49)
- Add `Reactions` feature (#671) (ba50af53)
- Add `ping_url` and `ping` to `Hook` (#669) (6169d8ea)
- Add `Repository.archived` property (#657) (35333e03)
- Add unit test for `tree` attribute of `GitCommit` (#668) (e5bfdbeb)
- Add `read_only` attribute to `Deploy Keys` (#570) (dbc6f5ab)
- Doc create instance from token (#667) (c33a3883)
- Fix uploading binary files on Python 3 (#621) (317079ef)
- Decode `jwt` bytes object in Python 3 (#633) (84b43da7)
- Remove broken downloads badge (#644) (15cdc2f8)
- Added missing parameters for repo creation (#623) (5c41120a)
- Add ability to access github Release Asset API. (#525) (52449649)

- Add ‘submitted at’ to PullRequestReview (#565) (ebe7277a)
- Quote path for /contents API (#614) (554c1ab1)
- Add Python 3.6 (2533bed9)
- Add Python 3.6 (e78f0ece)
- Updated references in introduction.rst (d2c72bb3)
- fix failing tests on py26 (291f6dde)
- Import missing exception (67b078e9)

3.1.6 Version 1.35 (July 10, 2017)

- Add Support for repository collaborator invitations.

3.1.7 Version 1.34 (abril 04, 2017)

- Add Support for Pull Request Reviews feature.

3.1.8 Version 1.32 (February 1, 2017)

- Support for Integrations installation endpoint (656e70e1)

3.1.9 Version 1.31 (January 30, 2017)

- Support HTTP 302 redirect in Organization.has_in_members (0154c6b)
- Add details of repo type for get_repos documentation (f119147)
- Note explicit support for Python 3.5 (3ae55f0)
- Fix README instructions (5b0224e)
- An easier to see link to the documentation in response to issue #480. (6039a4b)
- Encode GithubObject repr values in utf-8 when using Python2 (8ab9082)
- Updated documentation (4304ccd)
- Added a subscribers count field (a2da7f9)
- Added “add_to_assignees” & “remove_from_assignees” method to Issue object. (66430d7)
- Added “assignees” attribute to PullRequest object. (c0de6be)
- add html_url to GitRelease (ec633aa)
- Removed unused imports (65afc3f)
- Fix typo in a constant (10a28e0)
- Fix changelog formatting glitch (03a9227)
- Added “assignees” argument in Repository.create_issue() (ba007dc)
- Enhance support of “assignees” argument in Issue.edit() (14dd9f0)
- Added “assignees” attribute to Issue object. (e0e5fdf)

3.1.10 Version 1.30 (January 30, 2017)

- adds GitHub integrations (d60943d)

3.1.11 Version 1.29 (October 10, 2016)

- add issue assignee param (3a8edc7)
- Fix different case (fcf6cfb)
- DOC: remove easy_install suggestion; update links (45e76d9)
- Add permission param documentation (9347345)
- Add ability to set permission for team repo (5dddea7)
- Fix status check (073bb44)
- adds support for content dirs (0799753)

3.1.12 Version 1.28 (September 09, 2016)

- test against python 3.5 (5d35284)
- sort params and make them work on py3 (78374b9)
- adds a nicer `__repr__` (8571d87)
- Add missing space (464259d)
- Properly handle HTTP Proxy authentication with Python 3 (d015154)
- Fix small typo (987bca0)
- push to 'origin' instead of 'github' (d640666)

3.1.13 Version 1.27.1 (August 12, 2016)

- upgrade release process based on travis (3c20a66)
- change file content encoding to support unicode(like chinese), py2 (5404030)
- adds missing testfile corrections (9134aa2)
- fixed file API return values (0f29a53)
- assert by str and unicode to make it more py3 friendly (7390827)
- Patch issue 358 status context (#428) (70e30c5)
- Adding "since" param to Issue.get_comments() (#426) (3c6f99f)
- update doc url everywhere (#420) (cb0cf0a)
- fix a couple typos to be clearer (#419) (23c0e75)
- Document how one gets an AuthenticatedUser object (ba66862)
- fix wrong expectance on requestJsonAndCheck() returning {} if no data (8985368)
- Add previous_filename property to File (e1be1e6)
- add changelog entry for 1.26.0 (a1f3de2)

- update project files (be2e98b)
- fix update/create/delete file api return value issue (8bb765a)
- fix typo (a7929ac)
- fix update/delete/create content return value invalid issue (a0a4511)
- Follow redirects in the case of a 301 status code (c29f533)
- Fix for pickling exception when deserializing GithubException. (8f8b455)
- add support for the head parameter in Repository.get_pulls (397a74d)
- Add: - CommitCombinedStatus class - get_combined_status() to Commit class to return combined status - Add test for combined status. (5823ed7)
- fix python3 compatibility issue for using json/base64 (5b7f0bb)
- remove not covered API from readme (9c6f881)
- change replay data for update file test case (46895df)
- fix python3 compatability error in test case (00777db)
- Add repo content create/update/delete testcase (4aaeb9e)
- add MAINTAINERS file (a16b55b)
- travis: disable email (6347157)
- fix protect branch tests (65360b0)
- Add branch protection endpoint (737f0c3)
- fix request parameters issue (ae37d44)
- add content file create/update/delete api (b83ffbf)
- Add travis button on README. (a83649b)
- fix misspelling: <https://github.com/PyGithub/PyGithub/issues/363> (a06b5ec)
- Adding base parameter to get_pulls() method. (71593a8)
- add support for the direction parameter in Repository.get_pulls (70bcb6d)
- added creator parameter (ca9af4f)

3.1.14 Version 1.27.0 (August 12, 2016)

- this version was never released to PyPi due to a problem with the deployment

3.1.15 Version 1.26.0 (November 5th, 2015)

- Added context parameter to Status API
- Changed InputGitAuthor to reflect that time is an optional parameter
- Added sort option to get_pulls
- Added api_preview parameter to Requester class
- Return empty list instead of None for pagination with no pages
- Removed URL scheme validation that broke GitHub Enterprise

- Added “add_membership” call to Teams
- Added support to lazily load repositories
- Updated test suite to record with oauth tokens
- Added support for http_proxy
- Add support for filter/role options in Organization.get_members()
- Changed Organization.get_members’s filter parameter to _filter
- Fix escaping so that labels now support whitespaces
- Updated create_issue to support taking a list of strings for labels
- Added support for long integers in get_repo
- Fixed pagination to thread headers between requests
- Added repo.get_stargazers_with_dates()

3.1.16 Version 1.25.2 (October 7th, 2014)

- [Work around](#) the GitHub API v3 returning *nulls* in some paginated responses, [erichaase](#) for the bug report

3.1.17 Version 1.25.1 (September 28th, 2014)

- [Fix](#) two-factor authentication header, thanks to [tradej](#) for the pull request

3.1.18 Version 1.25.0 (May 4th, 2014)

- [Implement](#) getting repos by id, thanks to [tylertreat](#) for the pull request
- [Add](#) `Gist.owner`, thanks to [dalejung](#) for the pull request

3.1.19 Version 1.24.1 (March 16th, 2014)

- [Fix](#) urlquoting in search, thanks to [cro](#) for the pull request

3.1.20 Version 1.24.0 (March 2nd, 2014)

- [Implement](#) search, thanks to [thialfihar](#) for the pull request

3.1.21 Version 1.23.0 (December 23th, 2013)

- [Fix](#) all that is based on headers in Python 3 (pagination, conditional request, rate_limit...), huge thanks to [cwarren-mw](#) for finding the bug
- [Accept](#) strings for assignees and collaborators, thanks to [farrd](#)
- [Ease](#) two-factor authentication by adding ‘onetime_password’ to `AuthenticatedUser.create_authorization`, thanks to [cameronbwhite](#)

3.1.22 Version 1.22.0 (December 15th, 2013)

- Emojis, thanks to [evolvelight](#)
- `Repository.stargazers_count`, thanks to [cameronbwhite](#)
- `User.get_teams`, thanks to [poulp](#)

3.1.23 Version 1.21.0 (November ??th, 2013)

- **Accept** strings as well as `Label` objects in `Issue.add_to_labels`, `Issue.remove_from_labels` and `Issue.set_labels`. Thank you [acdha](#) for asking
- **Implement** equality comparison for *completable* github objects (ie. those who have a `url` attribute). Warning, comparison is still not implemented for non-completable objects. This will be done in version 2.0 of PyGithub. Thank you [OddBloke](#) for asking
- **Add** parameter `author` to `Repository.get_commits`. Thank you [naorosenberg](#) for asking
- **Implement** the statistics end points. Thank you [naorosenberg](#) for asking

3.1.24 Version 1.20.0 (October 20th, 2013) (First Seattle edition)

- **Implement** `Github.get_hook(name)`. Thank you [klmitch](#) for asking
- In case bad data is returned by Github API v3, **raise** an exception only when the user accesses the faulty attribute, not when constructing the object containing this attribute. Thank you [klmitch](#) for asking
- **Fix** parameter `public/private` of `Repository.edit`. Thank you [daireobroin449](#) for reporting the issue
- **Remove** `Repository.create_download` and `NamedUser.create_gist` as the corresponding APIs are not documented anymore

3.1.25 Version 1.19.0 (September 8th, 2013) (AKFish's edition)

- **Implement** [conditional requests](#) by the method `GithubObject.update`. Thank you very much [akfish](#) for the pull request and your collaboration!
- **Implement** persistence of PyGithub objects: `Github.save` and `Github.load`. Don't forget to update your objects after loading them, it won't decrease your rate limiting quota if nothing has changed. Again, thank you [akfish](#)
- **Implement** `Github.get_repos` to get all public repositories
- **Implement** `NamedUser.has_in_following`
- **Implement** `Github.get_api_status`, `Github.get_last_api_status_message` and `Github.get_api_status_messages`. Thank you [ruxandraburtica](#) for asking
- **Implement** `Github.get_rate_limit`
- Add many missing attributes
- **Technical change:** HTTP headers are now stored in retrieved objects. This is a base for new functionalities. Thank you [akfish](#) for the pull request
- Use the new URL to fork gists (minor change)
- Use the new URL to test hooks (minor change)

3.1.26 Version 1.18.0 (August 21st, 2013) (Bénodet edition)

- `Issues`’ `repository` attribute will never be `None`. Thank you [stuglaser](#) for the pull request
- No more false assumption on `rate_limiting`, and creation of `rate_limiting_resetime`. Thank you [edjackson](#) for the pull request
- New parameters `since` and `until` to `Repository.get_commits`. Thank you [apetresc](#) for the pull request
- Catch `Json` parsing exception for some internal server errors, and throw a better exception. Thank you [MarkRoddy](#) for the pull request
- Allow reversed iteration of `PaginatedList`. Thank you [davidbrai](#) for the pull request

3.1.27 Version 1.17.0 (July 7th, 2013) (Hamburg edition)

- Fix bug in `Repository.get_comment` when using custom `per_page`. Thank you [davidbrai](#)
- Handle `Http` redirects in `Repository.get_dir_contents`. Thank you [MarkRoddy](#)
- Implement `API /user` in `Github.get_users`. Thank you [rakeshcusat](#) for asking
- Improve the documentation. Thank you [martinqt](#)

3.1.28 Version 1.16.0 (May 31th, 2013) (Concarneau edition)

- Add the `html_url` attribute to `IssueComment` and `PullRequestComment`

3.1.29 Version 1.15.0 (May 17th, 2013) (Switzerland edition)

- Implement listing of user issues with all parameters. Thank you [Daehyok Shin](#) for reporting
- Raise two new specific exceptions

3.1.30 Version 1.14.2 (April 25th, 2013)

- Fix paginated requests when using `secret-key` `oauth`. Thank you [jseabold](#) for analysing the bug

3.1.31 Version 1.14.1 (April 25th, 2013)

- Set the default `User-Agent` header to “`PyGithub/Python`”. (`Github` has enforced the `User Agent` header yesterday.) Thank you [jjh42](#) for the fix, thank you [jasenmh](#) and [pconrad](#) for reporting the issue.

3.1.32 Version 1.14.0 (April 22nd, 2013)

- Improve `gist` edition. Thank you [jasonwiener](#) for asking:
 - Delete a file with `gist.edit(files={"name.txt": None})`
 - Rename a file with `gist.edit(files={"old_name.txt": github.InputFileContent(gist.files["old_name.txt"].content, new_name="new_name.txt")})`

- Raise specific exceptions. Thank you [pconrad](#) for giving me the idea

3.1.33 Version 1.13.1 (March 28nd, 2013)

- Fix login/password authentication for Python 3. Thank you [sebastianstigler](#) for reporting

3.1.34 Version 1.13.0 (March 22nd, 2013)

- Fix for Python 3 on case-insensitive file-systems. Thank you [ptwobrussell](#) for reporting
- Expose raw data returned by Github for all objects. Thank you [ptwobrussell](#) for asking
- Add a property `github.MainClass.Github.per_page` (and a parameter to the constructor) to change the number of items requested in paginated requests. Thank you again [ptwobrussell](#) for asking
- Implement the first part of the Notifications API. Thank you [pgolm](#)
- Fix automated tests on Python 3.3. Thank you [bkabrda](#) for reporting

3.1.35 Version 1.12.2 (March 3rd, 2013)

- Fix major issue with Python 3: Json decoding was broken. Thank you [bilderbuchi](#) for reporting

3.1.36 Version 1.12.1 (February 20th, 2013)

- Nothing, but packaging/upload of 1.12.0 failed

3.1.37 Version 1.12.0 (February 20th, 2013)

- Much better documentation: <http://jacquev6.github.com/PyGithub>
- Implement `github.Repository.Repository.get_dir_contents()`. Thank you [ksookocheff-va](#) for asking

3.1.38 Version 1.11.1 (February 9th, 2013) (London edition)

- Fix bug in lazy completion. Thank you [ianozsvald](#) for pinpointing it

3.1.39 Version 1.11.0 (February 7th, 2013)

- Fix bug in PaginatedList without url parameters. Thank you [lilimlib](#) for the contribution
- Implement `github.NamedUser.NamedUser.get_keys()`
- Support PubSubHub: `github.Repository.Repository.subscribe_to_hub()` and `github.Repository.Repository.unsubscribe_from_hub()`
- Publish the oauth scopes in `github.MainClass.Github.oauth_scopes`, thank you [bilderbuchi](#) for asking

3.1.40 Version 1.10.0 (December 25th, 2012) (Christmas 2012 edition)

- Major improvement: support Python 3! PyGithub is automatically tested on Travis with versions 2.5, 2.6, 2.7, 3.1 and 3.2 of Python
- Add a shortcut function `github.MainClass.Github.get_repo()` to get a repo directly from its full name. thank you [lwc](#) for the contribution
- `github.MainClass.Github.get_gitignore_templates()` and `github.MainClass.Github.get_gitignore_template()` for APIs /gitignore/templates
- Add the optional `ref` parameter to `github.Repository.Repository.get_contents()` and `github.Repository.Repository.get_readme()`. Thank you [fixxxeruk](#) for the contribution
- Get comments for all issues and all pull requests on a repository (GET /repos/:owner/:repo/pulls/comments: `github.Repository.Repository.get_pulls_comments()` or `github.Repository.Repository.get_pulls_review_comments()`; GET /repos/:owner/:repo/issues/comments: `github.Repository.Repository.get_issues_comments()`)

3.1.41 Version 1.9.1 (November 20th, 2012)

- Fix an assertion failure when integers returned by Github do not fit in a Python int

3.1.42 Version 1.9.0 (November 19th, 2012)

- You can now use your `client_id` and `client_secret` to increase rate limiting without authentication
- You can now send a custom User-Agent
- PullRequest now has its 'assignee' attribute, thank you [mstead](#)
- Repository.edit now has 'default_branch' parameter
- create_repo has 'auto_init' and 'gitignore_template' parameters
- GistComment URL is changed (see <http://developer.github.com/changes/2012-10-31-gist-comment-uris>)
- A typo in the readme was fixed by [tymofij](#), thank you
- Internal stuff:
 - Add encoding comment to Python files, thank you [Zearin](#)
 - Restore support of Python 2.5
 - Restore coverage measurement in setup.py test
 - Small refactoring

3.1.43 Version 1.8.1 (October 28th, 2012)

- `Repository.get_git_ref` prepends "refs/" to the requested references. Thank you [simon-weber](#) for noting the incoherence between documentation and behavior. If you feel like it's a breaking change, please see [this issue](#)

3.1.44 Version 1.8.0 (September 30th, 2012)

- Enable Travis CI
- Fix error 500 when json payload contains percent character (%). Thank you again [quixotique](#) for pointing that and reporting it to Github
- Enable debug logging. Logger name is “*github*”. Simple logging can be enabled by `github.enable_console_debug_logging()`. Thank you [quixotique](#) for the merge request and the advice
- Publish tests in the PyPi source archive to ease QA tests of the FreeBSD port. Thank you [koobs](#) for maintaining this port
- Switch to Semantic Versioning
- Respect pep8 Style Guide for Python Code

3.1.45 Version 1.7 (September 12th, 2012)

- Be able to clear the assignee and the milestone of an Issue. Thank you [quixotique](#) for the merge request
- Fix an AssertionError in `Organization.get_xxx` when using Github Enterprise. Thank you [mnsanghvi](#) for pointing that
- Expose pagination to users needing it (`PaginatedList.get_page`). Thank you [kukuts](#) for asking
- Improve handling of legacy search APIs
- Small refactoring (documentation, removal of old code generation artifacts)

3.1.46 Version 1.6 (September 8th, 2012)

- Restore support for Python 2.5
- Implement new APIS:
 - `/hooks` (undocumented, but mentioned in <http://developer.github.com/v3/repos/hooks/#create-a-hook>)
 - Merging
 - Starring and subscriptions
 - Assignees
 - Commit statuses
 - Contents, thank you [berndca](#) for asking
- Clarify issue and review comments on PullRequest, thank you [nixoz2k7](#) for asking

3.1.47 Version 1.5 (September 5th, 2012)

- Add a timeout option, thank you much [xobb1t](#) for the merge request. *This drops Python 2.5 support.* I may be able to restore it in next version.
- Implement `Repository.delete`, thank you [pmchen](#) for asking

3.1.48 Version 1.4 (August 4th, 2012)

- Allow connection to a custom Github URL, for Github Enterprise, thank you very much [engie](#) for the merge request

3.1.49 Version 1.3 (July 13th, 2012)

- Implement [markdown rendering](#)
- `GitAuthor.date` is now a datetime, thank you [bilderbuchi](#)
- Fix documentation of `Github.get_gist: id` is a string, not an integer

3.1.50 Version 1.2 (June 29th, 2012)

- Implement [legacy search APIs](#), thank you [kukuts](#) for telling me Github had released them
- Fix a bug with issue labels containing spaces, thank you [philipkimmey](#) for detecting the bug and fixing it
- Clarify how collections of objects are returned by `get_*` methods, thank you [bilderbuchi](#) for asking

3.1.51 Version 1.1 (June 20th, 2012)

- Restore compatibility with Python 2.5, thank you [pmuilu](#)
- Use `package_data` instead of `data_files` for documentation files in `setup.py`, thank you [malexw](#) for reporting

3.1.52 Version 1.0 (June 3rd, 2012)

- Complete rewrite, with no more complicated meta-description
- Full typing of attributes and parameters
- Full documentation of attributes and parameters
- More usable exceptions raised in case on problems with the API
- Some bugs and limitations fixed, special thanks to [bilderbuchi](#), [roskakori](#) and [tallforasmurf](#) for reporting them!

3.2 Pre-release versions

3.2.1 Version 0.7 (May 26th, 2012)

- Use PyGithub with OAuth authentication or with no authentication at all

3.2.2 Version 0.6 (April 17th, 2012)

- Fix [issue 21](#) (KeyError when accessing repositories)
- Re-completed the API with `NamedUser.create_gist`

3.2.3 Version 0.5 (March 19th, 2012)

- Major achievement: **all APIs are implemented**
- More refactoring, of course

3.2.4 Version 0.4 (March 12th, 2012)

- The list of the not implemented APIs is shorter than the list of the implemented APIs
- APIs *not implemented*:
 - GET */gists/public*
 - GET */issues*
 - GET */repos/:owner/:repo/compare/:base...:head*
 - GET */repos/:owner/:repo/git/trees/:sha?recursive-1*
 - POST */repos/:owner/:repo/git/trees?base_tree-*
- Gists
- Authorizations
- Keys
- Hooks
- Events
- Merge pull requests
- More refactoring, one more time

3.2.5 Version 0.3 (February 26th, 2012)

- More refactoring
- Issues, milestones and their labels
- NamedUser:
 - emails
- Repository:
 - downloads
 - tags, branches, commits and comments (not the same as “Git objects” of version 0.2)
 - pull requests (no automatic merge yet)
- Automatic generation of the reference documentation of classes, with less “see API”s, and less errors

3.2.6 Version 0.2 (February 23rd, 2012)

- Refactoring
- Teams details and modification
 - basic attributes

- list teams in organizations, on repositories
- Git objects
 - create and get tags, references, commits, trees, blobs
 - list and edit references

3.2.7 Version 0.1 (February 19th, 2012)

- User details and modification
 - basic attributes
 - followers, following, watching
 - organizations
 - repositories
- Repository details and modification
 - basic attributes
 - forking
 - collaborators, contributors, watchers
- Organization details and modification
 - basic attributes
 - members and public members

g

github, 5

github.GithubException, 24

A

- a (github.StatsContributor.StatsContributor.Week attribute), 106
- accept_invitation() (github.AuthenticatedUser.AuthenticatedUser method), 34
- accesskeyid (github.Download.Download attribute), 42
- acl (github.Download.Download attribute), 42
- active (github.Hook.Hook attribute), 55
- actor (github.Event.Event attribute), 43
- actor (github.IssueEvent.IssueEvent attribute), 62
- actual_value (github.GithubException.BadAttributeException attribute), 24
- add_membership() (github.Team.Team method), 109
- add_to_assignees() (github.Issue.Issue method), 59
- add_to_collaborators() (github.Repository.Repository method), 90
- add_to_emails() (github.AuthenticatedUser.AuthenticatedUser method), 28
- add_to_following() (github.AuthenticatedUser.AuthenticatedUser method), 28
- add_to_labels() (github.Issue.Issue method), 59
- add_to_labels() (github.PullRequest.PullRequest method), 81
- add_to_members() (github.Organization.Organization method), 71
- add_to_members() (github.Team.Team method), 109
- add_to_public_members() (github.Organization.Organization method), 71
- add_to_repos() (github.Team.Team method), 109
- add_to_starred() (github.AuthenticatedUser.AuthenticatedUser method), 28
- add_to_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 29
- add_to_watched() (github.AuthenticatedUser.AuthenticatedUser method), 29
- additions (github.CommitStats.CommitStats attribute), 39
- additions (github.File.File attribute), 44
- additions (github.PullRequest.PullRequest attribute), 76
- additions (github.StatsCodeFrequency.StatsCodeFrequency attribute), 106
- admin (github.Permissions.Permissions attribute), 76
- ahead_by (github.Comparison.Comparison attribute), 40
- all (github.StatsParticipation.StatsParticipation attribute), 107
- allow_merge_commit (github.Repository.Repository attribute), 85
- allow_rebase_merge (github.Repository.Repository attribute), 85
- allow_squash_merge (github.Repository.Repository attribute), 86
- app (github.Authorization.Authorization attribute), 34
- archive_url (github.Repository.Repository attribute), 86
- archived (github.Repository.Repository attribute), 86
- as_issue() (github.PullRequest.PullRequest method), 78
- as_pull_request() (github.Issue.Issue method), 59
- assignee (github.Issue.Issue attribute), 57
- assignee (github.PullRequest.PullRequest attribute), 76
- assignees (github.Issue.Issue attribute), 57
- assignees (github.PullRequest.PullRequest attribute), 76
- assignees_url (github.Repository.Repository attribute), 86
- AuthenticatedUser (class in github.AuthenticatedUser), 26
- author (github.Commit.Commit attribute), 36
- author (github.GitCommit.GitCommit attribute), 49
- author (github.GitRelease.GitRelease attribute), 51
- author (github.StatsContributor.StatsContributor attribute), 106
- Authorization (class in github.Authorization), 34
- AuthorizationApplication (class in github.AuthorizationApplication), 35
- authors_count (github.SourceImport.SourceImport attribute), 105
- authors_url (github.SourceImport.SourceImport attribute), 105
- avatar_url (github.AuthenticatedUser.AuthenticatedUser attribute), 26
- avatar_url (github.NamedUser.NamedUser attribute), 65
- avatar_url (github.Organization.Organization attribute), 70

B

- BadAttributeException, 24
 - BadCredentialsException, 24
 - BadUserAgentException, 24
 - base (github.PullRequest.PullRequest attribute), 76
 - base_commit (github.Comparison.Comparison attribute), 40
 - behind_by (github.Comparison.Comparison attribute), 40
 - billing_email (github.Organization.Organization attribute), 70
 - bio (github.AuthenticatedUser.AuthenticatedUser attribute), 26
 - bio (github.NamedUser.NamedUser attribute), 65
 - blob_url (github.File.File attribute), 44
 - blobs_url (github.Repository.Repository attribute), 86
 - blog (github.AuthenticatedUser.AuthenticatedUser attribute), 26
 - blog (github.NamedUser.NamedUser attribute), 65
 - blog (github.Organization.Organization attribute), 70
 - body (github.CommitComment.CommitComment attribute), 38
 - body (github.GistComment.GistComment attribute), 46
 - body (github.GitRelease.GitRelease attribute), 51
 - body (github.Issue.Issue attribute), 57
 - body (github.IssueComment.IssueComment attribute), 61
 - body (github.License.License attribute), 63
 - body (github.PullRequest.PullRequest attribute), 76
 - body (github.PullRequestComment.PullRequestComment attribute), 82
 - body (github.PullRequestReview.PullRequestReview attribute), 84
 - body (github.StatusMessage.StatusMessage attribute), 107
 - Branch (class in github.Branch), 36
 - branches_url (github.Repository.Repository attribute), 86
 - browser_download_url (github.GitReleaseAsset.GitReleaseAsset attribute), 53
 - bucket (github.Download.Download attribute), 42
- C**
- c (github.StatsContributor.StatsContributor.Week attribute), 106
 - change_status (github.GistHistoryState.GistHistoryState attribute), 47
 - changed_files (github.PullRequest.PullRequest attribute), 77
 - changes (github.File.File attribute), 44
 - clone_url (github.Repository.Repository attribute), 86
 - closed_at (github.Issue.Issue attribute), 58
 - closed_at (github.PullRequest.PullRequest attribute), 77
 - closed_by (github.Issue.Issue attribute), 58
 - closed_issues (github.Milestone.Milestone attribute), 64
 - code (github.HookResponse.HookResponse attribute), 56
 - collaborators (github.AuthenticatedUser.AuthenticatedUser attribute), 26
 - collaborators (github.NamedUser.NamedUser attribute), 65
 - collaborators (github.Organization.Organization attribute), 70
 - collaborators (github.Plan.Plan attribute), 76
 - collaborators_url (github.Repository.Repository attribute), 86
 - color (github.Label.Label attribute), 62
 - comments (github.Gist.Gist attribute), 44
 - comments (github.GistHistoryState.GistHistoryState attribute), 48
 - comments (github.Issue.Issue attribute), 58
 - comments (github.PullRequest.PullRequest attribute), 77
 - comments_url (github.Commit.Commit attribute), 36
 - comments_url (github.Gist.Gist attribute), 44
 - comments_url (github.GistHistoryState.GistHistoryState attribute), 48
 - comments_url (github.Issue.Issue attribute), 58
 - comments_url (github.PullRequest.PullRequest attribute), 77
 - comments_url (github.Repository.Repository attribute), 86
 - Commit (class in github.Commit), 36
 - commit (github.Branch.Branch attribute), 36
 - commit (github.Commit.Commit attribute), 36
 - commit (github.Tag.Tag attribute), 108
 - commit_id (github.CommitComment.CommitComment attribute), 38
 - commit_id (github.IssueEvent.IssueEvent attribute), 62
 - commit_id (github.PullRequestComment.PullRequestComment attribute), 82
 - commit_id (github.PullRequestReview.PullRequestReview attribute), 84
 - commit_url (github.CommitCombinedStatus.CommitCombinedStatus attribute), 38
 - CommitCombinedStatus (class in github.CommitCombinedStatus), 37
 - CommitComment (class in github.CommitComment), 38
 - commits (github.Comparison.Comparison attribute), 40
 - commits (github.PullRequest.PullRequest attribute), 77
 - commits_url (github.Gist.Gist attribute), 44
 - commits_url (github.GistHistoryState.GistHistoryState attribute), 48
 - commits_url (github.PullRequest.PullRequest attribute), 77
 - commits_url (github.Repository.Repository attribute), 86
 - CommitStats (class in github.CommitStats), 39
 - CommitStatus (class in github.CommitStatus), 39
 - committed_at (github.GistHistoryState.GistHistoryState attribute), 48
 - committer (github.Commit.Commit attribute), 36
 - committer (github.GitCommit.GitCommit attribute), 49

- company (github.AuthenticatedUser.AuthenticatedUser attribute), 26
- company (github.NamedUser.NamedUser attribute), 65
- company (github.Organization.Organization attribute), 70
- compare() (github.Repository.Repository method), 90
- compare_url (github.Repository.Repository attribute), 86
- Comparison (class in github.Comparison), 40
- conditions (github.License.License attribute), 63
- config (github.Hook.Hook attribute), 55
- content (github.ContentFile.ContentFile attribute), 41
- content (github.GistFile.GistFile attribute), 47
- content (github.GitBlob.GitBlob attribute), 49
- content (github.Reaction.Reaction attribute), 85
- content_type (github.Download.Download attribute), 42
- content_type (github.GitReleaseAsset.GitReleaseAsset attribute), 52
- ContentFile (class in github.ContentFile), 41
- contents_url (github.File.File attribute), 44
- contents_url (github.Repository.Repository attribute), 86
- context (github.CommitStatus.CommitStatus attribute), 40
- contexts (github.Branch.Branch attribute), 36
- contributors_url (github.Repository.Repository attribute), 86
- convert_to_outside_collaborator() (github.Organization.Organization method), 74
- create_authorization() (github.AuthenticatedUser.AuthenticatedUser method), 29
- create_comment() (github.Commit.Commit method), 37
- create_comment() (github.Gist.Gist method), 45
- create_comment() (github.Issue.Issue method), 59
- create_comment() (github.PullRequest.PullRequest method), 78
- create_file() (github.Repository.Repository method), 95
- create_fork() (github.AuthenticatedUser.AuthenticatedUser method), 29
- create_fork() (github.Gist.Gist method), 45
- create_fork() (github.Organization.Organization method), 72
- create_from_raw_data() (github.MainClass.Github method), 9
- create_gist() (github.AuthenticatedUser.AuthenticatedUser method), 29
- create_git_blob() (github.Repository.Repository method), 90
- create_git_commit() (github.Repository.Repository method), 90
- create_git_ref() (github.Repository.Repository method), 91
- create_git_release() (github.Repository.Repository method), 91
- create_git_tag() (github.Repository.Repository method), 91
- create_git_tree() (github.Repository.Repository method), 91
- create_hook() (github.Organization.Organization method), 72
- create_hook() (github.Repository.Repository method), 91
- create_issue() (github.Repository.Repository method), 92
- create_issue_comment() (github.PullRequest.PullRequest method), 79
- create_key() (github.AuthenticatedUser.AuthenticatedUser method), 29
- create_key() (github.Repository.Repository method), 92
- create_label() (github.Repository.Repository method), 92
- create_milestone() (github.Repository.Repository method), 92
- create_pull() (github.Repository.Repository method), 93
- create_reaction() (github.CommitComment.CommitComment method), 39
- create_reaction() (github.Issue.Issue method), 60
- create_reaction() (github.IssueComment.IssueComment method), 61
- create_reaction() (github.PullRequestComment.PullRequestComment method), 83
- create_repo() (github.AuthenticatedUser.AuthenticatedUser method), 30
- create_repo() (github.Organization.Organization method), 72
- create_review() (github.PullRequest.PullRequest method), 79
- create_review_comment() (github.PullRequest.PullRequest method), 79
- create_review_request() (github.PullRequest.PullRequest method), 79
- create_source_import() (github.Repository.Repository method), 93
- create_status() (github.Commit.Commit method), 37
- create_team() (github.Organization.Organization method), 72
- created_at (github.AuthenticatedUser.AuthenticatedUser attribute), 27
- created_at (github.Authorization.Authorization attribute), 35
- created_at (github.CommitComment.CommitComment attribute), 38
- created_at (github.CommitStatus.CommitStatus attribute), 39
- created_at (github.Download.Download attribute), 42
- created_at (github.Event.Event attribute), 43
- created_at (github.Gist.Gist attribute), 44
- created_at (github.GistComment.GistComment attribute), 46
- created_at (github.GistHistoryState.GistHistoryState attribute), 48
- created_at (github.GitRelease.GitRelease attribute), 51
- created_at (github.GitReleaseAsset.GitReleaseAsset at-

- tribute), 53
 - created_at (github.Hook.Hook attribute), 55
 - created_at (github.Invitation.Invitation attribute), 57
 - created_at (github.Issue.Issue attribute), 58
 - created_at (github.IssueComment.IssueComment attribute), 61
 - created_at (github.IssueEvent.IssueEvent attribute), 62
 - created_at (github.Milestone.Milestone attribute), 64
 - created_at (github.NamedUser.NamedUser attribute), 65
 - created_at (github.Organization.Organization attribute), 70
 - created_at (github.PullRequest.PullRequest attribute), 77
 - created_at (github.PullRequestComment.PullRequestComment attribute), 82
 - created_at (github.Reaction.Reaction attribute), 85
 - created_at (github.Repository.Repository attribute), 86
 - created_at (github.RepositoryKey.RepositoryKey attribute), 104
 - created_on (github.StatusMessage.StatusMessage attribute), 108
 - creator (github.CommitStatus.CommitStatus attribute), 39
 - creator (github.Milestone.Milestone attribute), 64
- ## D
- d (github.StatsContributor.StatsContributor.Week attribute), 106
 - data (github.GithubException.GithubException attribute), 24
 - date (github.GitAuthor.GitAuthor attribute), 49
 - days (github.StatsCommitActivity.StatsCommitActivity attribute), 106
 - default_branch (github.Repository.Repository attribute), 86
 - delete() (github.Authorization.Authorization method), 35
 - delete() (github.CommitComment.CommitComment method), 39
 - delete() (github.Download.Download method), 43
 - delete() (github.Gist.Gist method), 46
 - delete() (github.GistComment.GistComment method), 47
 - delete() (github.GitRef.GitRef method), 50
 - delete() (github.Hook.Hook method), 55
 - delete() (github.IssueComment.IssueComment method), 61
 - delete() (github.Label.Label method), 63
 - delete() (github.Milestone.Milestone method), 64
 - delete() (github.PullRequestComment.PullRequestComment method), 83
 - delete() (github.Reaction.Reaction method), 85
 - delete() (github.Repository.Repository method), 93
 - delete() (github.RepositoryKey.RepositoryKey method), 104
 - delete() (github.Team.Team method), 109
 - delete() (github.UserKey.UserKey method), 111
 - delete_asset() (github.GitReleaseAsset.GitReleaseAsset method), 53
 - delete_file() (github.Repository.Repository method), 96
 - delete_hook() (github.Organization.Organization method), 73
 - delete_labels() (github.Issue.Issue method), 59
 - delete_labels() (github.PullRequest.PullRequest method), 81
 - delete_release() (github.GitRelease.GitRelease method), 52
 - delete_review_request() (github.PullRequest.PullRequest method), 79
 - deletions (github.CommitStats.CommitStats attribute), 39
 - deletions (github.File.File attribute), 44
 - deletions (github.PullRequest.PullRequest attribute), 77
 - deletions (github.StatsCodeFrequency.StatsCodeFrequency attribute), 106
 - description (github.CommitStatus.CommitStatus attribute), 39
 - description (github.Download.Download attribute), 42
 - description (github.Gist.Gist attribute), 45
 - description (github.GistHistoryState.GistHistoryState attribute), 48
 - description (github.Label.Label attribute), 62
 - description (github.License.License attribute), 63
 - description (github.Milestone.Milestone attribute), 64
 - description (github.Organization.Organization attribute), 70
 - description (github.Repository.Repository attribute), 86
 - description (github.Team.Team attribute), 108
 - diff_hunk (github.PullRequestComment.PullRequestComment attribute), 82
 - diff_url (github.Comparison.Comparison attribute), 40
 - diff_url (github.IssuePullRequest.IssuePullRequest attribute), 62
 - diff_url (github.PullRequest.PullRequest attribute), 77
 - disk_usage (github.AuthenticatedUser.AuthenticatedUser attribute), 27
 - disk_usage (github.NamedUser.NamedUser attribute), 65
 - disk_usage (github.Organization.Organization attribute), 70
 - Download (class in github.Download), 42
 - download_count (github.Download.Download attribute), 42
 - download_count (github.GitReleaseAsset.GitReleaseAsset attribute), 53
 - download_url (github.ContentFile.ContentFile attribute), 41
 - downloads_url (github.Repository.Repository attribute), 86
 - draft (github.GitRelease.GitRelease attribute), 51
 - due_on (github.Milestone.Milestone attribute), 64
 - dump() (github.MainClass.Github method), 9

E

edit() (github.AuthenticatedUser.AuthenticatedUser method), 30

edit() (github.Authorization.Authorization method), 35

edit() (github.CommitComment.CommitComment method), 39

edit() (github.Gist.Gist method), 46

edit() (github.GistComment.GistComment method), 47

edit() (github.GitRef.GitRef method), 50

edit() (github.Hook.Hook method), 55

edit() (github.Issue.Issue method), 59

edit() (github.IssueComment.IssueComment method), 61

edit() (github.Label.Label method), 63

edit() (github.Milestone.Milestone method), 65

edit() (github.Organization.Organization method), 73

edit() (github.PullRequest.PullRequest method), 79

edit() (github.PullRequestComment.PullRequestComment method), 83

edit() (github.Repository.Repository method), 93

edit() (github.Team.Team method), 109

edit_hook() (github.Organization.Organization method), 73

email (github.AuthenticatedUser.AuthenticatedUser attribute), 27

email (github.GitAuthor.GitAuthor attribute), 49

email (github.NamedUser.NamedUser attribute), 65

email (github.Organization.Organization attribute), 70

enable_console_debug_logging() (in module github), 24

encoding (github.ContentFile.ContentFile attribute), 41

encoding (github.GitBlob.GitBlob attribute), 49

enforcement_level (github.Branch.Branch attribute), 36

etag (github.GithubObject.GithubObject attribute), 26

Event (class in github.Event), 43

event (github.IssueEvent.IssueEvent attribute), 62

events (github.Hook.Hook attribute), 55

events (github.HookDescription.HookDescription attribute), 56

events_url (github.AuthenticatedUser.AuthenticatedUser attribute), 27

events_url (github.Issue.Issue attribute), 58

events_url (github.NamedUser.NamedUser attribute), 65

events_url (github.Organization.Organization attribute), 70

events_url (github.Repository.Repository attribute), 86

expected_type (github.GithubException.BadAttributeException attribute), 24

expirationdate (github.Download.Download attribute), 42

expires_at (github.InstallationAuthorization.InstallationAuthorization attribute), 57

F

File (class in github.File), 44

filename (github.File.File attribute), 44

filename (github.GistFile.GistFile attribute), 47

files (github.Commit.Commit attribute), 36

files (github.Comparison.Comparison attribute), 40

files (github.Gist.Gist attribute), 45

files (github.GistHistoryState.GistHistoryState attribute), 48

FIX_REPO_GET_GIT_REF (github.MainClass.Github attribute), 5

followers (github.AuthenticatedUser.AuthenticatedUser attribute), 27

followers (github.NamedUser.NamedUser attribute), 65

followers (github.Organization.Organization attribute), 70

followers_url (github.AuthenticatedUser.AuthenticatedUser attribute), 27

followers_url (github.NamedUser.NamedUser attribute), 65

following (github.AuthenticatedUser.AuthenticatedUser attribute), 27

following (github.NamedUser.NamedUser attribute), 66

following (github.Organization.Organization attribute), 70

following_url (github.AuthenticatedUser.AuthenticatedUser attribute), 27

following_url (github.NamedUser.NamedUser attribute), 66

fork (github.Repository.Repository attribute), 87

fork_of (github.Gist.Gist attribute), 45

forks (github.Gist.Gist attribute), 45

forks (github.GistHistoryState.GistHistoryState attribute), 48

forks (github.Repository.Repository attribute), 87

forks_count (github.Repository.Repository attribute), 87

forks_url (github.Gist.Gist attribute), 45

forks_url (github.GistHistoryState.GistHistoryState attribute), 48

forks_url (github.Repository.Repository attribute), 87

full_name (github.Repository.Repository attribute), 87

G

get() (github.StatsPunchCard.StatsPunchCard method), 107

get_repr__() (github.GithubObject.GithubObject method), 26

get_api_status() (github.MainClass.Github method), 9

get_api_status_messages() (github.MainClass.Github method), 9

get_archive_link() (github.Repository.Repository method), 94

get_assignees() (github.GitRelease.GitRelease method), 52

get_assignees() (github.Repository.Repository method), 94

get_authorization() (github.AuthenticatedUser.AuthenticatedUser method), 30

get_authorizations() (github.AuthenticatedUser.AuthenticatedUser method), 31

- [get_branch\(\)](#) (github.Repository.Repository method), 94
[get_branches\(\)](#) (github.Repository.Repository method), 94
[get_collaborators\(\)](#) (github.Repository.Repository method), 94
[get_combined_status\(\)](#) (github.Commit.Commit method), 37
[get_comment\(\)](#) (github.Gist.Gist method), 46
[get_comment\(\)](#) (github.Issue.Issue method), 59
[get_comment\(\)](#) (github.PullRequest.PullRequest method), 80
[get_comment\(\)](#) (github.Repository.Repository method), 94
[get_comments\(\)](#) (github.Commit.Commit method), 37
[get_comments\(\)](#) (github.Gist.Gist method), 46
[get_comments\(\)](#) (github.Issue.Issue method), 60
[get_comments\(\)](#) (github.PullRequest.PullRequest method), 80
[get_comments\(\)](#) (github.Repository.Repository method), 94
[get_commit\(\)](#) (github.Repository.Repository method), 95
[get_commits\(\)](#) (github.PullRequest.PullRequest method), 80
[get_commits\(\)](#) (github.Repository.Repository method), 95
[get_contents\(\)](#) (github.Repository.Repository method), 95
[get_contributors\(\)](#) (github.Repository.Repository method), 97
[get_dir_contents\(\)](#) (github.Repository.Repository method), 96
[get_download\(\)](#) (github.Repository.Repository method), 97
[get_downloads\(\)](#) (github.Repository.Repository method), 97
[get_emails\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 31
[get_emojis\(\)](#) (github.MainClass.Github method), 9
[get_events\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 31
[get_events\(\)](#) (github.Issue.Issue method), 60
[get_events\(\)](#) (github.NamedUser.NamedUser method), 67
[get_events\(\)](#) (github.Organization.Organization method), 73
[get_events\(\)](#) (github.Repository.Repository method), 97
[get_file_contents\(\)](#) (github.Repository.Repository method), 95
[get_files\(\)](#) (github.PullRequest.PullRequest method), 80
[get_followers\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 31
[get_followers\(\)](#) (github.NamedUser.NamedUser method), 67
[get_following\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 31
[get_following\(\)](#) (github.NamedUser.NamedUser method), 67
[get_forks\(\)](#) (github.Repository.Repository method), 97
[get_gist\(\)](#) (github.MainClass.Github method), 7
[get_gists\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 31
[get_gists\(\)](#) (github.MainClass.Github method), 7
[get_gists\(\)](#) (github.NamedUser.NamedUser method), 67
[get_git_blob\(\)](#) (github.Repository.Repository method), 97
[get_git_commit\(\)](#) (github.Repository.Repository method), 97
[get_git_ref\(\)](#) (github.Repository.Repository method), 97
[get_git_refs\(\)](#) (github.Repository.Repository method), 97
[get_git_tag\(\)](#) (github.Repository.Repository method), 98
[get_git_tree\(\)](#) (github.Repository.Repository method), 98
[get_gitignore_template\(\)](#) (github.MainClass.Github method), 9
[get_gitignore_templates\(\)](#) (github.MainClass.Github method), 9
[get_hook\(\)](#) (github.MainClass.Github method), 8
[get_hook\(\)](#) (github.Organization.Organization method), 73
[get_hook\(\)](#) (github.Repository.Repository method), 98
[get_hooks\(\)](#) (github.MainClass.Github method), 8
[get_hooks\(\)](#) (github.Organization.Organization method), 74
[get_hooks\(\)](#) (github.Repository.Repository method), 98
[get_installation\(\)](#) (github.MainClass.Github method), 10
[get_issue\(\)](#) (github.Repository.Repository method), 98
[get_issue_comment\(\)](#) (github.PullRequest.PullRequest method), 80
[get_issue_comments\(\)](#) (github.PullRequest.PullRequest method), 81
[get_issues\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 31
[get_issues\(\)](#) (github.Organization.Organization method), 74
[get_issues\(\)](#) (github.Repository.Repository method), 98
[get_issues_comments\(\)](#) (github.Repository.Repository method), 99
[get_issues_event\(\)](#) (github.Repository.Repository method), 99
[get_issues_events\(\)](#) (github.Repository.Repository method), 99
[get_key\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 32
[get_key\(\)](#) (github.Repository.Repository method), 99
[get_keys\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 32
[get_keys\(\)](#) (github.NamedUser.NamedUser method), 67
[get_keys\(\)](#) (github.Repository.Repository method), 99
[get_label\(\)](#) (github.Repository.Repository method), 99
[get_labels\(\)](#) (github.Issue.Issue method), 60
[get_labels\(\)](#) (github.Milestone.Milestone method), 65
[get_labels\(\)](#) (github.PullRequest.PullRequest method), 81

- [get_labels\(\) \(github.Repository.Repository method\), 99](#)
[get_languages\(\) \(github.Repository.Repository method\), 99](#)
[get_last_api_status_message\(\) \(github.MainClass.Github method\), 9](#)
[get_latest_release\(\) \(github.Repository.Repository method\), 102](#)
[get_license\(\) \(github.MainClass.Github method\), 6](#)
[get_license\(\) \(github.Repository.Repository method\), 99](#)
[get_licenses\(\) \(github.MainClass.Github method\), 6](#)
[get_members\(\) \(github.Organization.Organization method\), 74](#)
[get_members\(\) \(github.Team.Team method\), 110](#)
[get_milestone\(\) \(github.Repository.Repository method\), 100](#)
[get_milestones\(\) \(github.Repository.Repository method\), 100](#)
[get_network_events\(\) \(github.Repository.Repository method\), 100](#)
[get_notification\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 32](#)
[get_notifications\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 32](#)
[get_organization\(\) \(github.MainClass.Github method\), 6](#)
[get_organization_events\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 32](#)
[get_organizations\(\) \(github.MainClass.Github method\), 6](#)
[get_orgs\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 32](#)
[get_orgs\(\) \(github.NamedUser.NamedUser method\), 67](#)
[get_outside_collaborators\(\) \(github.Organization.Organization method\), 74](#)
[get_protected_branch\(\) \(github.Repository.Repository method\), 94](#)
[get_public_events\(\) \(github.NamedUser.NamedUser method\), 68](#)
[get_public_members\(\) \(github.Organization.Organization method\), 74](#)
[get_public_received_events\(\) \(github.NamedUser.NamedUser method\), 68](#)
[get_pull\(\) \(github.Repository.Repository method\), 100](#)
[get_pulls\(\) \(github.Repository.Repository method\), 100](#)
[get_pulls_comments\(\) \(github.Repository.Repository method\), 100](#)
[get_pulls_review_comments\(\) \(github.Repository.Repository method\), 101](#)
[get_rate_limit\(\) \(github.MainClass.Github method\), 6](#)
[get_reactions\(\) \(github.CommitComment.CommitComment method\), 39](#)
[get_reactions\(\) \(github.Issue.Issue method\), 60](#)
[get_reactions\(\) \(github.IssueComment.IssueComment method\), 61](#)
[get_reactions\(\) \(github.PullRequestComment.PullRequestComment method\), 83](#)
[get_readme\(\) \(github.Repository.Repository method\), 101](#)
[get_received_events\(\) \(github.NamedUser.NamedUser method\), 68](#)
[get_release\(\) \(github.Repository.Repository method\), 102](#)
[get_releases\(\) \(github.Repository.Repository method\), 102](#)
[get_repo\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 32](#)
[get_repo\(\) \(github.MainClass.Github method\), 7](#)
[get_repo\(\) \(github.NamedUser.NamedUser method\), 68](#)
[get_repo\(\) \(github.Organization.Organization method\), 75](#)
[get_repos\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 33](#)
[get_repos\(\) \(github.Installation.Installation method\), 56](#)
[get_repos\(\) \(github.MainClass.Github method\), 7](#)
[get_repos\(\) \(github.NamedUser.NamedUser method\), 68](#)
[get_repos\(\) \(github.Organization.Organization method\), 75](#)
[get_repos\(\) \(github.Team.Team method\), 110](#)
[get_review\(\) \(github.PullRequest.PullRequest method\), 81](#)
[get_review_comment\(\) \(github.PullRequest.PullRequest method\), 80](#)
[get_review_comments\(\) \(github.PullRequest.PullRequest method\), 80](#)
[get_review_requests\(\) \(github.PullRequest.PullRequest method\), 81](#)
[get_reviews\(\) \(github.PullRequest.PullRequest method\), 81](#)
[get_single_review_comments\(\) \(github.PullRequest.PullRequest method\), 80](#)
[get_source_import\(\) \(github.Repository.Repository method\), 101](#)
[get_stargazers\(\) \(github.Repository.Repository method\), 101](#)
[get_stargazers_with_dates\(\) \(github.Repository.Repository method\), 101](#)
[get_starred\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 33](#)
[get_starred\(\) \(github.NamedUser.NamedUser method\), 68](#)
[get_starred_gists\(\) \(github.AuthenticatedUser.AuthenticatedUser method\), 33](#)
[get_stats_code_frequency\(\) \(github.Repository.Repository method\), 101](#)
[get_stats_commit_activity\(\) \(github.Repository.Repository method\), 101](#)
[get_stats_contributors\(\) \(github.Repository.Repository method\), 101](#)

- get_stats_participation() (github.Repository.Repository method), 101
 - get_stats_punch_card() (github.Repository.Repository method), 102
 - get_status() (github.GitRef.GitRef method), 51
 - get_statuses() (github.Commit.Commit method), 37
 - get_statuses() (github.GitRef.GitRef method), 51
 - get_subscribers() (github.Repository.Repository method), 102
 - get_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 33
 - get_subscriptions() (github.NamedUser.NamedUser method), 68
 - get_tags() (github.Repository.Repository method), 102
 - get_team() (github.Organization.Organization method), 75
 - get_teams() (github.AuthenticatedUser.AuthenticatedUser method), 33
 - get_teams() (github.Organization.Organization method), 75
 - get_teams() (github.Repository.Repository method), 102
 - get_topics() (github.Repository.Repository method), 102
 - get_user() (github.MainClass.Github method), 6
 - get_user_issues() (github.AuthenticatedUser.AuthenticatedUser method), 31
 - get_users() (github.MainClass.Github method), 6
 - get_watched() (github.AuthenticatedUser.AuthenticatedUser method), 33
 - get_watched() (github.NamedUser.NamedUser method), 68
 - get_watchers() (github.Repository.Repository method), 102
 - Gist (class in github.Gist), 44
 - GistComment (class in github.GistComment), 46
 - GistFile (class in github.GistFile), 47
 - GistHistoryState (class in github.GistHistoryState), 47
 - gists_url (github.AuthenticatedUser.AuthenticatedUser attribute), 27
 - gists_url (github.NamedUser.NamedUser attribute), 66
 - git_commits_url (github.Repository.Repository attribute), 87
 - git_pull_url (github.Gist.Gist attribute), 45
 - git_pull_url (github.GistHistoryState.GistHistoryState attribute), 48
 - git_push_url (github.Gist.Gist attribute), 45
 - git_push_url (github.GistHistoryState.GistHistoryState attribute), 48
 - git_refs_url (github.Repository.Repository attribute), 87
 - git_tags_url (github.Repository.Repository attribute), 87
 - git_url (github.ContentFile.ContentFile attribute), 41
 - git_url (github.Repository.Repository attribute), 87
 - GitAuthor (class in github.GitAuthor), 49
 - GitBlob (class in github.GitBlob), 49
 - GitCommit (class in github.GitCommit), 49
 - Github (class in github.MainClass), 5
 - github (module), 5
 - github.GithubException (module), 24
 - GithubException, 24
 - GithubObject (class in github.GithubObject), 26
 - GitignoreTemplate (class in github.GitignoreTemplate), 54
 - GitObject (class in github.GitObject), 50
 - GitRef (class in github.GitRef), 50
 - GitRelease (class in github.GitRelease), 51
 - GitReleaseAsset (class in github.GitReleaseAsset), 52
 - GitTag (class in github.GitTag), 53
 - GitTree (class in github.GitTree), 54
 - GitTreeElement (class in github.GitTreeElement), 54
 - gravatar_id (github.AuthenticatedUser.AuthenticatedUser attribute), 27
 - gravatar_id (github.NamedUser.NamedUser attribute), 66
 - gravatar_id (github.Organization.Organization attribute), 70
- ## H
- has_downloads (github.Repository.Repository attribute), 87
 - has_in_assignees() (github.Repository.Repository method), 102
 - has_in_collaborators() (github.Repository.Repository method), 102
 - has_in_following() (github.AuthenticatedUser.AuthenticatedUser method), 33
 - has_in_following() (github.NamedUser.NamedUser method), 68
 - has_in_members() (github.Organization.Organization method), 75
 - has_in_members() (github.Team.Team method), 110
 - has_in_public_members() (github.Organization.Organization method), 75
 - has_in_repos() (github.Team.Team method), 110
 - has_in_starred() (github.AuthenticatedUser.AuthenticatedUser method), 33
 - has_in_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 33
 - has_in_watched() (github.AuthenticatedUser.AuthenticatedUser method), 34
 - has_issues (github.Repository.Repository attribute), 87
 - has_large_files (github.SourceImport.SourceImport attribute), 105
 - has_projects (github.Repository.Repository attribute), 87
 - has_wiki (github.Repository.Repository attribute), 87
 - head (github.PullRequest.PullRequest attribute), 77
 - hireable (github.AuthenticatedUser.AuthenticatedUser attribute), 27
 - hireable (github.NamedUser.NamedUser attribute), 66
 - history (github.Gist.Gist attribute), 45

- history (github.GistHistoryState.GistHistoryState attribute), 48
- homepage (github.Repository.Repository attribute), 87
- Hook (class in github.Hook), 55
- HookDescription (class in github.HookDescription), 56
- HookResponse (class in github.HookResponse), 56
- hooks_url (github.Repository.Repository attribute), 87
- html_url (github.AuthenticatedUser.AuthenticatedUser attribute), 27
- html_url (github.Commit.Commit attribute), 36
- html_url (github.CommitComment.CommitComment attribute), 38
- html_url (github.Comparison.Comparison attribute), 40
- html_url (github.ContentFile.ContentFile attribute), 41
- html_url (github.Download.Download attribute), 42
- html_url (github.Gist.Gist attribute), 45
- html_url (github.GistHistoryState.GistHistoryState attribute), 48
- html_url (github.GitCommit.GitCommit attribute), 50
- html_url (github.GitRelease.GitRelease attribute), 52
- html_url (github.Invitation.Invitation attribute), 57
- html_url (github.Issue.Issue attribute), 58
- html_url (github.IssueComment.IssueComment attribute), 61
- html_url (github.IssuePullRequest.IssuePullRequest attribute), 62
- html_url (github.License.License attribute), 63
- html_url (github.NamedUser.NamedUser attribute), 66
- html_url (github.Organization.Organization attribute), 70
- html_url (github.PullRequest.PullRequest attribute), 77
- html_url (github.PullRequestComment.PullRequestComment attribute), 83
- html_url (github.PullRequestReview.PullRequestReview attribute), 84
- html_url (github.Repository.Repository attribute), 87
- html_url (github.SourceImport.SourceImport attribute), 105
- I**
- id (github.AuthenticatedUser.AuthenticatedUser attribute), 27
- id (github.Authorization.Authorization attribute), 35
- id (github.CommitComment.CommitComment attribute), 38
- id (github.CommitStatus.CommitStatus attribute), 40
- id (github.Download.Download attribute), 42
- id (github.Event.Event attribute), 43
- id (github.Gist.Gist attribute), 45
- id (github.GistComment.GistComment attribute), 46
- id (github.GistHistoryState.GistHistoryState attribute), 48
- id (github.GitRelease.GitRelease attribute), 51
- id (github.GitReleaseAsset.GitReleaseAsset attribute), 52
- id (github.Hook.Hook attribute), 55
- id (github.Invitation.Invitation attribute), 57
- id (github.Issue.Issue attribute), 58
- id (github.IssueComment.IssueComment attribute), 61
- id (github.IssueEvent.IssueEvent attribute), 62
- id (github.Milestone.Milestone attribute), 64
- id (github.NamedUser.NamedUser attribute), 66
- id (github.Notification.Notification attribute), 69
- id (github.Organization.Organization attribute), 70
- id (github.PullRequest.PullRequest attribute), 77
- id (github.PullRequestComment.PullRequestComment attribute), 82
- id (github.PullRequestReview.PullRequestReview attribute), 84
- id (github.Reaction.Reaction attribute), 85
- id (github.Repository.Repository attribute), 87
- id (github.RepositoryKey.RepositoryKey attribute), 104
- id (github.Team.Team attribute), 108
- id (github.UserKey.UserKey attribute), 111
- implementation (github.License.License attribute), 63
- in_reply_to_id (github.PullRequestComment.PullRequestComment attribute), 82
- InputFileContent (class in github.InputFileContent), 25
- InputGitAuthor (class in github.InputGitAuthor), 25
- InputGitTreeElement (class in github.InputGitTreeElement), 25
- Installation (class in github.Installation), 56
- InstallationAuthorization (class in github.InstallationAuthorization), 57
- Invitation (class in github.Invitation), 57
- is_merged() (github.PullRequest.PullRequest method), 82
- is_starred() (github.Gist.Gist method), 46
- Issue (class in github.Issue), 57
- issue (github.IssueEvent.IssueEvent attribute), 62
- issue_comment_url (github.Repository.Repository attribute), 87
- issue_events_url (github.Repository.Repository attribute), 88
- issue_url (github.IssueComment.IssueComment attribute), 61
- issue_url (github.PullRequest.PullRequest attribute), 77
- IssueComment (class in github.IssueComment), 61
- IssueEvent (class in github.IssueEvent), 62
- IssuePullRequest (class in github.IssuePullRequest), 62
- issues_url (github.Repository.Repository attribute), 88
- K**
- key (github.License.License attribute), 63
- key (github.RepositoryKey.RepositoryKey attribute), 104
- key (github.UserKey.UserKey attribute), 111
- keys_url (github.Repository.Repository attribute), 88
- L**
- Label (class in github.Label), 62

- label (github.GitReleaseAsset.GitReleaseAsset attribute), 52
 - label (github.PullRequestPart.PullRequestPart attribute), 84
 - labels (github.Issue.Issue attribute), 58
 - labels (github.PullRequest.PullRequest attribute), 77
 - labels_url (github.Issue.Issue attribute), 58
 - labels_url (github.Milestone.Milestone attribute), 64
 - labels_url (github.Repository.Repository attribute), 88
 - language (github.GistFile.GistFile attribute), 47
 - language (github.Repository.Repository attribute), 88
 - languages_url (github.Repository.Repository attribute), 88
 - large_files_count (github.SourceImport.SourceImport attribute), 105
 - large_files_size (github.SourceImport.SourceImport attribute), 105
 - last_modified (github.GithubObject.GithubObject attribute), 26
 - last_read_at (github.Notification.Notification attribute), 69
 - last_response (github.Hook.Hook attribute), 55
 - last_updated (github.Status.Status attribute), 107
 - latest_comment_url (github.NotificationSubject.NotificationSubject attribute), 69
 - legacy_search_issues() (github.Repository.Repository method), 103
 - License (class in github.License), 63
 - license (github.ContentFile.ContentFile attribute), 41
 - limit (github.Rate.Rate attribute), 85
 - limitations (github.License.License attribute), 64
 - line (github.CommitComment.CommitComment attribute), 38
 - load() (github.MainClass.Github method), 9
 - location (github.AuthenticatedUser.AuthenticatedUser attribute), 27
 - location (github.NamedUser.NamedUser attribute), 66
 - location (github.Organization.Organization attribute), 70
 - login (github.AuthenticatedUser.AuthenticatedUser attribute), 27
 - login (github.NamedUser.NamedUser attribute), 66
 - login (github.Organization.Organization attribute), 70
- ## M
- master_branch (github.Repository.Repository attribute), 88
 - members_count (github.Team.Team attribute), 108
 - members_url (github.Organization.Organization attribute), 71
 - members_url (github.Team.Team attribute), 108
 - merge() (github.PullRequest.PullRequest method), 82
 - merge() (github.Repository.Repository method), 103
 - merge_base_commit (github.Comparison.Comparison attribute), 40
 - merge_commit_sha (github.PullRequest.PullRequest attribute), 77
 - mergeable (github.PullRequest.PullRequest attribute), 77
 - mergeable_state (github.PullRequest.PullRequest attribute), 77
 - merged (github.PullRequest.PullRequest attribute), 77
 - merged (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 83
 - merged_at (github.PullRequest.PullRequest attribute), 78
 - merged_by (github.PullRequest.PullRequest attribute), 78
 - merges_url (github.Repository.Repository attribute), 88
 - message (github.GitCommit.GitCommit attribute), 50
 - message (github.GitTag.GitTag attribute), 53
 - message (github.HookResponse.HookResponse attribute), 56
 - message (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 83
 - Milestone (class in github.Milestone), 64
 - milestone (github.Issue.Issue attribute), 58
 - milestone (github.PullRequest.PullRequest attribute), 78
 - milestones_url (github.Repository.Repository attribute), 88
 - mime_type (github.Download.Download attribute), 42
 - Subject (class in github.Repository.Repository attribute), 88
 - mode (github.GitTreeElement.GitTreeElement attribute), 54
- ## N
- name (github.AuthenticatedUser.AuthenticatedUser attribute), 27
 - name (github.AuthorizationApplication.AuthorizationApplication attribute), 35
 - name (github.Branch.Branch attribute), 36
 - name (github.ContentFile.ContentFile attribute), 41
 - name (github.Download.Download attribute), 42
 - name (github.GitAuthor.GitAuthor attribute), 49
 - name (github.GitignoreTemplate.GitignoreTemplate attribute), 54
 - name (github.GitReleaseAsset.GitReleaseAsset attribute), 52
 - name (github.Hook.Hook attribute), 55
 - name (github.HookDescription.HookDescription attribute), 56
 - name (github.Label.Label attribute), 62
 - name (github.License.License attribute), 63
 - name (github.NamedUser.NamedUser attribute), 66
 - name (github.Organization.Organization attribute), 71
 - name (github.Plan.Plan attribute), 76
 - name (github.Repository.Repository attribute), 88
 - name (github.Tag.Tag attribute), 108
 - name (github.Team.Team attribute), 108
 - NamedUser (class in github.NamedUser), 65
 - network_count (github.Repository.Repository attribute), 88

- note (github.Authorization.Authorization attribute), 35
- note_url (github.Authorization.Authorization attribute), 35
- Notification (class in github.Notification), 69
- notifications_url (github.Repository.Repository attribute), 88
- NotificationSubject (class in github.NotificationSubject), 69
- number (github.Issue.Issue attribute), 58
- number (github.Milestone.Milestone attribute), 64
- number (github.PullRequest.PullRequest attribute), 78
- ## O
- oauth_scopes (github.MainClass.Github attribute), 6
- object (github.GitRef.GitRef attribute), 50
- object (github.GitTag.GitTag attribute), 53
- on_behalf_of (github.InstallationAuthorization.InstallationAuthorization attribute), 57
- open_issues (github.Milestone.Milestone attribute), 64
- open_issues (github.Repository.Repository attribute), 88
- open_issues_count (github.Repository.Repository attribute), 88
- org (github.Event.Event attribute), 43
- Organization (class in github.Organization), 70
- organization (github.Repository.Repository attribute), 88
- organization (github.Team.Team attribute), 109
- organizations_url (github.AuthenticatedUser.AuthenticatedUser attribute), 27
- organizations_url (github.NamedUser.NamedUser attribute), 66
- original_commit_id (github.PullRequestComment.PullRequestComment attribute), 82
- original_position (github.PullRequestComment.PullRequestComment attribute), 82
- owned_private_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 27
- owned_private_repos (github.NamedUser.NamedUser attribute), 66
- owned_private_repos (github.Organization.Organization attribute), 71
- owner (github.Gist.Gist attribute), 45
- owner (github.GistHistoryState.GistHistoryState attribute), 48
- owner (github.Repository.Repository attribute), 88
- owner (github.StatsParticipation.StatsParticipation attribute), 107
- ## P
- PaginatedList (class in github.PaginatedList), 25
- parent (github.Repository.Repository attribute), 88
- parents (github.Commit.Commit attribute), 36
- parents (github.GitCommit.GitCommit attribute), 50
- patch (github.File.File attribute), 44
- patch_url (github.Comparison.Comparison attribute), 40
- patch_url (github.IssuePullRequest.IssuePullRequest attribute), 62
- patch_url (github.PullRequest.PullRequest attribute), 78
- path (github.CommitComment.CommitComment attribute), 38
- path (github.ContentFile.ContentFile attribute), 41
- path (github.Download.Download attribute), 42
- path (github.GitTreeElement.GitTreeElement attribute), 54
- path (github.PullRequestComment.PullRequestComment attribute), 82
- payload (github.Event.Event attribute), 43
- per_page (github.MainClass.Github attribute), 6
- permalink_url (github.Comparison.Comparison attribute), 40
- permission (github.Team.Team attribute), 108
- Permissions (class in github.Permissions), 76
- permissions (github.Invitation.Invitation attribute), 57
- permissions (github.License.License attribute), 63
- permissions (github.NamedUser.NamedUser attribute), 66
- permissions (github.Repository.Repository attribute), 89
- ping() (github.Hook.Hook method), 56
- ping_url (github.Hook.Hook attribute), 55
- Plan (class in github.Plan), 76
- plan (github.AuthenticatedUser.AuthenticatedUser attribute), 28
- plan (github.NamedUser.NamedUser attribute), 66
- plan (github.Organization.Organization attribute), 71
- policy (github.Download.Download attribute), 42
- position (github.CommitComment.CommitComment attribute), 38
- position (github.PullRequestComment.PullRequestComment attribute), 82
- prefix (github.Download.Download attribute), 42
- prerelease (github.GitRelease.GitRelease attribute), 51
- previous_filename (github.File.File attribute), 44
- privacy (github.Team.Team attribute), 109
- private (github.Repository.Repository attribute), 89
- private_gists (github.AuthenticatedUser.AuthenticatedUser attribute), 28
- private_gists (github.NamedUser.NamedUser attribute), 66
- private_gists (github.Organization.Organization attribute), 71
- private_repos (github.Plan.Plan attribute), 76
- protect_branch() (github.Repository.Repository method), 103
- protected (github.Branch.Branch attribute), 36
- public (github.Event.Event attribute), 43
- public (github.Gist.Gist attribute), 45
- public (github.GistHistoryState.GistHistoryState attribute), 48

- public_gists (github.AuthenticatedUser.AuthenticatedUser attribute), 28
- public_gists (github.NamedUser.NamedUser attribute), 66
- public_gists (github.Organization.Organization attribute), 71
- public_members_url (github.Organization.Organization attribute), 71
- public_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 28
- public_repos (github.NamedUser.NamedUser attribute), 66
- public_repos (github.Organization.Organization attribute), 71
- published_at (github.GitRelease.GitRelease attribute), 51
- pull (github.Permissions.Permissions attribute), 76
- pull_request (github.Issue.Issue attribute), 58
- pull_request_url (github.PullRequestComment.PullRequestComment attribute), 82
- pull_request_url (github.PullRequestReview.PullRequestReview attribute), 84
- PullRequest (class in github.PullRequest), 76
- PullRequestComment (class in github.PullRequestComment), 82
- PullRequestMergeStatus (class in github.PullRequestMergeStatus), 83
- PullRequestPart (class in github.PullRequestPart), 84
- PullRequestReview (class in github.PullRequestReview), 84
- pulls_url (github.Repository.Repository attribute), 89
- push (github.Permissions.Permissions attribute), 76
- pushed_at (github.Repository.Repository attribute), 89
- ## R
- Rate (class in github.Rate), 85
- rate (github.RateLimit.RateLimit attribute), 85
- rate_limiting (github.MainClass.Github attribute), 6
- rate_limiting_resetime (github.MainClass.Github attribute), 6
- RateLimit (class in github.RateLimit), 85
- RateLimitExceededException, 24
- raw_data (github.GithubObject.GithubObject attribute), 26
- raw_headers (github.GithubObject.GithubObject attribute), 26
- raw_url (github.File.File attribute), 44
- raw_url (github.GistFile.GistFile attribute), 47
- Reaction (class in github.Reaction), 85
- read_only (github.RepositoryKey.RepositoryKey attribute), 104
- reason (github.Notification.Notification attribute), 69
- received_events_url (github.AuthenticatedUser.AuthenticatedUser attribute), 28
- received_events_url (github.NamedUser.NamedUser attribute), 66
- redirect (github.Download.Download attribute), 43
- ref (github.GitRef.GitRef attribute), 50
- ref (github.PullRequestPart.PullRequestPart attribute), 84
- remaining (github.Rate.Rate attribute), 85
- remove_from_assignees() (github.Issue.Issue method), 60
- remove_from_collaborators() (github.Repository.Repository method), 103
- remove_from_emails() (github.AuthenticatedUser.AuthenticatedUser method), 34
- remove_from_following() (github.AuthenticatedUser.AuthenticatedUser method), 34
- remove_from_labels() (github.Issue.Issue method), 60
- remove_from_labels() (github.PullRequest.PullRequest method), 81
- remove_from_members() (github.Organization.Organization method), 75
- remove_from_members() (github.Team.Team method), 110
- remove_from_membership() (github.Organization.Organization method), 75
- remove_from_public_members() (github.Organization.Organization method), 75
- remove_from_repos() (github.Team.Team method), 110
- remove_from_starred() (github.AuthenticatedUser.AuthenticatedUser method), 34
- remove_from_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 34
- remove_from_watched() (github.AuthenticatedUser.AuthenticatedUser method), 34
- remove_membership() (github.Team.Team method), 110
- remove_outside_collaborator() (github.Organization.Organization method), 74
- render_markdown() (github.MainClass.Github method), 8
- replace_topics() (github.Repository.Repository method), 103
- repo (github.Event.Event attribute), 43
- repo (github.PullRequestPart.PullRequestPart attribute), 84
- repos_count (github.Team.Team attribute), 108
- repos_url (github.AuthenticatedUser.AuthenticatedUser attribute), 28
- repos_url (github.NamedUser.NamedUser attribute), 67
- repos_url (github.Organization.Organization attribute), 71
- repositories_url (github.Team.Team attribute), 108
- Repository (class in github.Repository), 85
- repository (github.CommitCombinedStatus.CommitCombinedStatus attribute), 38
- repository (github.ContentFile.ContentFile attribute), 41

- repository (github.Invitation.Invitation attribute), 57
 - repository (github.Issue.Issue attribute), 58
 - repository (github.Notification.Notification attribute), 69
 - repository_url (github.SourceImport.SourceImport attribute), 105
 - RepositoryKey (class in github.RepositoryKey), 104
 - reset (github.Rate.Rate attribute), 85
 - reset_starred() (github.Gist.Gist method), 46
 - review_comment_url (github.PullRequest.PullRequest attribute), 78
 - review_comments (github.PullRequest.PullRequest attribute), 78
 - review_comments_url (github.PullRequest.PullRequest attribute), 78
- ## S
- s3_url (github.Download.Download attribute), 43
 - schema (github.HookDescription.HookDescription attribute), 56
 - scopes (github.Authorization.Authorization attribute), 35
 - search_code() (github.MainClass.Github method), 8
 - search_commits() (github.MainClass.Github method), 8
 - search_issues() (github.MainClass.Github method), 7
 - search_repositories() (github.MainClass.Github method), 7
 - search_users() (github.MainClass.Github method), 7
 - set_labels() (github.Issue.Issue method), 60
 - set_labels() (github.PullRequest.PullRequest method), 81
 - set_repo_permission() (github.Team.Team method), 109
 - set_starred() (github.Gist.Gist method), 46
 - sha (github.Commit.Commit attribute), 36
 - sha (github.CommitCombinedStatus.CommitCombinedStatus attribute), 37
 - sha (github.ContentFile.ContentFile attribute), 41
 - sha (github.File.File attribute), 44
 - sha (github.GitBlob.GitBlob attribute), 49
 - sha (github.GitCommit.GitCommit attribute), 50
 - sha (github.GitObject.GitObject attribute), 50
 - sha (github.GitTag.GitTag attribute), 53
 - sha (github.GitTree.GitTree attribute), 54
 - sha (github.GitTreeElement.GitTreeElement attribute), 54
 - sha (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 83
 - sha (github.PullRequestPart.PullRequestPart attribute), 84
 - signature (github.Download.Download attribute), 43
 - site_admin (github.AuthenticatedUser.AuthenticatedUser attribute), 28
 - site_admin (github.NamedUser.NamedUser attribute), 67
 - size (github.ContentFile.ContentFile attribute), 41
 - size (github.Download.Download attribute), 43
 - size (github.GistFile.GistFile attribute), 47
 - size (github.GitBlob.GitBlob attribute), 49
 - size (github.GitReleaseAsset.GitReleaseAsset attribute), 53
 - size (github.GitTreeElement.GitTreeElement attribute), 54
 - size (github.Repository.Repository attribute), 89
 - slug (github.Team.Team attribute), 108
 - source (github.GitignoreTemplate.GitignoreTemplate attribute), 54
 - source (github.Repository.Repository attribute), 89
 - SourceImport (class in github.SourceImport), 105
 - space (github.Plan.Plan attribute), 76
 - spdx_id (github.License.License attribute), 63
 - ssh_url (github.Repository.Repository attribute), 89
 - Stargazer (class in github.Stargazer), 105
 - stargazers_count (github.Repository.Repository attribute), 89
 - stargazers_url (github.Repository.Repository attribute), 89
 - starred_at (github.Stargazer.Stargazer attribute), 105
 - starred_url (github.AuthenticatedUser.AuthenticatedUser attribute), 28
 - starred_url (github.NamedUser.NamedUser attribute), 67
 - state (github.CommitCombinedStatus.CommitCombinedStatus attribute), 37
 - state (github.CommitStatus.CommitStatus attribute), 40
 - state (github.GitReleaseAsset.GitReleaseAsset attribute), 52
 - state (github.Issue.Issue attribute), 58
 - state (github.Milestone.Milestone attribute), 64
 - state (github.PullRequest.PullRequest attribute), 78
 - state (github.PullRequestReview.PullRequestReview attribute), 84
 - stats (github.Commit.Commit attribute), 36
 - StatsCodeFrequency (class in github.StatsCodeFrequency), 106
 - StatsCommitActivity (class in github.StatsCommitActivity), 106
 - StatsContributor (class in github.StatsContributor), 106
 - StatsContributor.Week (class in github.StatsContributor), 106
 - StatsParticipation (class in github.StatsParticipation), 107
 - StatsPunchCard (class in github.StatsPunchCard), 107
 - Status (class in github.Status), 107
 - status (github.Comparison.Comparison attribute), 41
 - status (github.File.File attribute), 44
 - status (github.GithubException.GithubException attribute), 24
 - status (github.HookResponse.HookResponse attribute), 56
 - status (github.SourceImport.SourceImport attribute), 105
 - status (github.Status.Status attribute), 107
 - status (github.StatusMessage.StatusMessage attribute), 107
 - status_text (github.SourceImport.SourceImport attribute),

- 105
 - statuses (github.CommitCombinedStatus.CommitCombinedStatus attribute), 38
 - statuses_url (github.Repository.Repository attribute), 89
 - StatusMessage (class in github.StatusMessage), 107
 - subject (github.Notification.Notification attribute), 69
 - submitted_at (github.PullRequestReview.PullRequestReview attribute), 84
 - subscribe_to_hub() (github.Repository.Repository method), 103
 - subscribers_count (github.Repository.Repository attribute), 89
 - subscribers_url (github.Repository.Repository attribute), 89
 - subscription_url (github.Notification.Notification attribute), 69
 - subscription_url (github.Repository.Repository attribute), 89
 - subscriptions_url (github.AuthenticatedUser.AuthenticatedUser attribute), 28
 - subscriptions_url (github.NamedUser.NamedUser attribute), 67
 - supported_events (github.HookDescription.HookDescription attribute), 56
 - svn_url (github.Repository.Repository attribute), 89
- ## T
- Tag (class in github.Tag), 108
 - tag (github.GitTag.GitTag attribute), 53
 - tag_name (github.GitRelease.GitRelease attribute), 51
 - tagger (github.GitTag.GitTag attribute), 53
 - tags_url (github.Repository.Repository attribute), 89
 - tarball_url (github.GitRelease.GitRelease attribute), 52
 - tarball_url (github.Tag.Tag attribute), 108
 - target_commitish (github.GitRelease.GitRelease attribute), 51
 - target_url (github.CommitStatus.CommitStatus attribute), 40
 - Team (class in github.Team), 108
 - teams_url (github.Repository.Repository attribute), 89
 - test() (github.Hook.Hook method), 56
 - test_url (github.Hook.Hook attribute), 55
 - title (github.GitRelease.GitRelease attribute), 51
 - title (github.Issue.Issue attribute), 58
 - title (github.Milestone.Milestone attribute), 64
 - title (github.NotificationSubject.NotificationSubject attribute), 69
 - title (github.PullRequest.PullRequest attribute), 78
 - title (github.RepositoryKey.RepositoryKey attribute), 104
 - title (github.UserKey.UserKey attribute), 111
 - token (github.Authorization.Authorization attribute), 35
 - token (github.InstallationAuthorization.InstallationAuthorization attribute), 57
 - topics (github.Repository.Repository attribute), 89
 - total (github.CommitStats.CommitStats attribute), 39
 - total (github.StatsCommitActivity.StatsCommitActivity attribute), 106
 - total (github.StatsContributor.StatsContributor attribute), 107
 - total_commits (github.Comparison.Comparison attribute), 41
 - total_count (github.CommitCombinedStatus.CommitCombinedStatus attribute), 38
 - total_private_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 28
 - total_private_repos (github.NamedUser.NamedUser attribute), 67
 - total_private_repos (github.Organization.Organization attribute), 71
 - transformation_exception (github.GithubException.BadAttributeException attribute), 24
 - tree (github.GitCommit.GitCommit attribute), 50
 - tree (github.GitTree.GitTree attribute), 54
 - trees_url (github.Repository.Repository attribute), 89
 - TwoFactorException, 24
 - type (github.AuthenticatedUser.AuthenticatedUser attribute), 28
 - type (github.ContentFile.ContentFile attribute), 41
 - type (github.Event.Event attribute), 43
 - type (github.GistFile.GistFile attribute), 47
 - type (github.GitObject.GitObject attribute), 50
 - type (github.GitTreeElement.GitTreeElement attribute), 54
 - type (github.NamedUser.NamedUser attribute), 67
 - type (github.NotificationSubject.NotificationSubject attribute), 69
 - type (github.Organization.Organization attribute), 71
- ## U
- UnknownObjectException, 24
 - unread (github.Notification.Notification attribute), 69
 - unsubscribe_from_hub() (github.Repository.Repository method), 104
 - update_asset() (github.GitReleaseAsset.GitReleaseAsset method), 53
 - update_file() (github.Repository.Repository method), 96
 - update_release() (github.GitRelease.GitRelease method), 52
 - updated_at (github.AuthenticatedUser.AuthenticatedUser attribute), 28
 - updated_at (github.Authorization.Authorization attribute), 35
 - updated_at (github.CommitComment.CommitComment attribute), 38
 - updated_at (github.CommitStatus.CommitStatus attribute), 40
 - updated_at (github.Gist.Gist attribute), 45

- updated_at (github.GistComment.GistComment attribute), 47
 - updated_at (github.GistHistoryState.GistHistoryState attribute), 48
 - updated_at (github.GitReleaseAsset.GitReleaseAsset attribute), 53
 - updated_at (github.Hook.Hook attribute), 55
 - updated_at (github.Issue.Issue attribute), 58
 - updated_at (github.IssueComment.IssueComment attribute), 61
 - updated_at (github.Milestone.Milestone attribute), 64
 - updated_at (github.NamedUser.NamedUser attribute), 67
 - updated_at (github.Notification.Notification attribute), 69
 - updated_at (github.Organization.Organization attribute), 71
 - updated_at (github.PullRequest.PullRequest attribute), 78
 - updated_at (github.PullRequestComment.PullRequestComment attribute), 82
 - updated_at (github.Repository.Repository attribute), 90
 - upload_asset() (github.GitRelease.GitRelease method), 52
 - upload_url (github.GitRelease.GitRelease attribute), 52
 - uploader (github.GitReleaseAsset.GitReleaseAsset attribute), 53
 - url (github.AuthenticatedUser.AuthenticatedUser attribute), 28
 - url (github.Authorization.Authorization attribute), 35
 - url (github.AuthorizationApplication.AuthorizationApplication attribute), 35
 - url (github.Commit.Commit attribute), 37
 - url (github.CommitCombinedStatus.CommitCombinedStatus attribute), 38
 - url (github.CommitComment.CommitComment attribute), 38
 - url (github.CommitStatus.CommitStatus attribute), 40
 - url (github.Comparison.Comparison attribute), 41
 - url (github.ContentFile.ContentFile attribute), 42
 - url (github.Download.Download attribute), 43
 - url (github.Gist.Gist attribute), 45
 - url (github.GistComment.GistComment attribute), 47
 - url (github.GistHistoryState.GistHistoryState attribute), 48
 - url (github.GitBlob.GitBlob attribute), 49
 - url (github.GitCommit.GitCommit attribute), 50
 - url (github.GitObject.GitObject attribute), 50
 - url (github.GitRef.GitRef attribute), 50
 - url (github.GitRelease.GitRelease attribute), 51
 - url (github.GitReleaseAsset.GitReleaseAsset attribute), 52
 - url (github.GitTag.GitTag attribute), 53
 - url (github.GitTree.GitTree attribute), 54
 - url (github.GitTreeElement.GitTreeElement attribute), 54
 - url (github.Hook.Hook attribute), 55
 - url (github.Invitation.Invitation attribute), 57
 - url (github.Issue.Issue attribute), 58
 - url (github.IssueComment.IssueComment attribute), 61
 - url (github.IssueEvent.IssueEvent attribute), 62
 - url (github.Label.Label attribute), 63
 - url (github.License.License attribute), 63
 - url (github.Milestone.Milestone attribute), 64
 - url (github.NamedUser.NamedUser attribute), 67
 - url (github.Notification.Notification attribute), 69
 - url (github.NotificationSubject.NotificationSubject attribute), 69
 - url (github.Organization.Organization attribute), 71
 - url (github.PullRequest.PullRequest attribute), 78
 - url (github.PullRequestComment.PullRequestComment attribute), 83
 - url (github.PullRequestReview.PullRequestReview attribute), 84
 - url (github.Repository.Repository attribute), 90
 - url (github.RepositoryKey.RepositoryKey attribute), 104
 - url (github.SourceImport.SourceImport attribute), 105
 - url (github.Team.Team attribute), 109
 - url (github.UserKey.UserKey attribute), 111
 - use_ifs (github.SourceImport.SourceImport attribute), 105
 - user (github.CommitComment.CommitComment attribute), 39
 - user (github.Gist.Gist attribute), 45
 - user (github.GistComment.GistComment attribute), 47
 - user (github.GistHistoryState.GistHistoryState attribute), 49
 - user (github.Issue.Issue attribute), 59
 - user (github.IssueComment.IssueComment attribute), 61
 - user (github.PullRequest.PullRequest attribute), 78
 - user (github.PullRequestComment.PullRequestComment attribute), 83
 - user (github.PullRequestPart.PullRequestPart attribute), 84
 - user (github.PullRequestReview.PullRequestReview attribute), 84
 - user (github.Reaction.Reaction attribute), 85
 - user (github.Stargazer.Stargazer attribute), 105
 - UserKey (class in github.UserKey), 111
- ## V
- vcs (github.SourceImport.SourceImport attribute), 105
 - vcs_url (github.SourceImport.SourceImport attribute), 105
 - verified (github.RepositoryKey.RepositoryKey attribute), 104
 - verified (github.UserKey.UserKey attribute), 111
 - version (github.GistHistoryState.GistHistoryState attribute), 49
- ## W
- w (github.StatsContributor.StatsContributor.Week attribute), 49

tribute), 106
watchers (github.Repository.Repository attribute), 90
watchers_count (github.Repository.Repository attribute),
90
week (github.StatsCodeFrequency.StatsCodeFrequency
attribute), 106
week (github.StatsCommitActivity.StatsCommitActivity
attribute), 106
weeks (github.StatsContributor.StatsContributor at-
tribute), 107

Z

zipball_url (github.GitRelease.GitRelease attribute), 52
zipball_url (github.Tag.Tag attribute), 108