

---

# PyGithub Documentation

*Release 1.38*

**Vincent Jacques**

**Mar 21, 2018**



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Reference</b>	<b>3</b>
<b>3</b>	<b>Change log</b>	<b>103</b>
	<b>Python Module Index</b>	<b>117</b>



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/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

## 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](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=10, client_id=None,
                                client_secret=None, user_agent='PyGithub/Python',
                                per_page=30, api_preview=False)
```

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

**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\_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\_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 informations 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()`
- /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()`
- /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/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.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()`
  - POST: `github.Issue.Issue.add_to_labels()`
  - PUT: `github.Issue.Issue.set_labels()`
  - DELETE: `github.Issue.Issue.delete_labels()`
- /repos/:owner/:repo/issues/:number/labels/:name
  - DELETE: `github.Issue.Issue.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/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.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_reviewer_requests()`
- /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()`
- /repos/:owner/:repo/releases/:id
  - GET: `github.Repository.Repository.get_release()`
- /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/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()`
- `/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()`
  - PATCH: `github.UserKey.UserKey.edit()`
  - 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()`

## 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)
```

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**

**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\_authorizations** ()

**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 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

**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  
<<https://developer.github.com/v3/reactions/#list-reactions-for-a-pull-request-review-comment>>  
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*

### 2.4.3 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

**encoding**  
Type string

**git\_url**  
Type string

**html\_url**  
Type string

**name**

**Type** string  
**path**  
**Type** string  
**repository**  
**Type** *github.Repository.Repository*  
**sha**  
**Type** string  
**size**  
**Type** integer  
**type**  
**Type** string  
**url**  
**Type** string

## 2.4.4 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*

## 2.4.5 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*

**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** `label` – `github.Label.Label`

**Return type** None

**get\_reactions()**

**Calls** `GET /repos/:owner/:repo/issues/:number/reactions <https://developer.github.com/v3/reactions/#list-reactions-for-an-issue>`

**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`

## 2.4.6 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

**permission**

**Type** string

**repos\_count**

**Type** integer

**repositories\_url**

**Type** string

**slug**

**Type** string

**url**

**Type** string

**organization**

**Type** *github.Organization.Organization*

**add\_to\_members** (*member*)

**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*, *permission=NotSet*)

**Calls** PATCH /teams/:id

**Parameters**

- **name** – string
- **permission** – 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\_from\_members** (*member*)

**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

## 2.4.7 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

## 2.4.8 AuthorizationApplication

**class** `github.AuthorizationApplication.AuthorizationApplication`

This class represents AuthorizationApplications

**name**

**Type** string

**url**

**Type** string

## 2.4.9 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

### 2.4.10 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

## 2.4.11 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

## 2.4.12 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

### 2.4.13 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  
**edit** (*title=NotSet, key=NotSet*)  
**Calls** `PATCH /user/keys/:id`  
**Parameters**

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

**Return type** None

### 2.4.14 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

### 2.4.15 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

### 2.4.16 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

**name**

**Type** string

**url**



**Type** string

**delete** ()

**Calls** `DELETE /repos/:owner/:repo/labels/:name`

**Return type** None

**edit** (*name, color*)

**Calls** `PATCH /repos/:owner/:repo/labels/:name`

**Parameters**

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

**Return type** None

### 2.4.17 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

### 2.4.18 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`

### 2.4.19 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

### 2.4.20 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

### 2.4.21 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

## 2.4.22 PullRequestReviewerRequest

**class** github.PullRequestReviewerRequest.**PullRequestReviewerRequest**

This class represents PullRequestReviewerRequests. The reference can be found here [https://developer.github.com/v3/pulls/review\\_requests/](https://developer.github.com/v3/pulls/review_requests/)

**login**

**Type** string

**id**

**Type** integer

## 2.4.23 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

**contributions**

**Type** integer

**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

## 2.4.24 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`

## 2.4.25 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`

## 2.4.26 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

### 2.4.27 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

### 2.4.28 HookResponse

**class** `github.HookResponse.HookResponse`

This class represents HookResponses

**code**  
Type integer

**message**  
Type string

**status**  
Type string

### 2.4.29 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*

### 2.4.30 IssuePullRequest

**class** `github.IssuePullRequest.IssuePullRequest`

This class represents IssuePullRequests

**diff\_url**

**Type** string

**html\_url**

**Type** string

**patch\_url**

**Type** string

### 2.4.31 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`

**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\_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*)

**Calls** `POST /orgs/:org/teams`

**Parameters**

- `name` – string
- `repo_names` – list of `github.Repository.Repository`
- `permission` – 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*, *email=NotSet*, *location=NotSet*, *name=NotSet*)

**Calls** `PATCH /orgs/:org`

**Parameters**

- `billing_email` – string
- `blog` – string
- `company` – 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\_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\_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

## 2.4.32 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

### 2.4.33 GitAuthor

**class** github.GitAuthor.**GitAuthor**

This class represents GitAuthors

**date**  
    **Type** datetime.datetime  
**email**  
    **Type** string  
**name**  
    **Type** string

### 2.4.34 Status

**class** github.Status.**Status**

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

**status**  
    **Type** string  
**last\_updated**  
    **Type** datetime.datetime

### 2.4.35 Repository

**class** github.Repository.**Repository**

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

**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\_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  
**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\_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*)

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

**Parameters**

- **name** – string
- **color** – 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*, *\*\*kwargs*)

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 https://developer.github.com/v3/migration/source_imports/#start-an-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\_wiki=NotSet*, *has\_downloads=NotSet*, *default\_branch=NotSet*)

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

**Parameters**

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



- **private** – bool
- **has\_issues** – bool
- **has\_wiki** – bool
- **has\_downloads** – bool
- **default\_branch** – string

**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` `:param path: string, (required), path of the file in the repository` `:param message: string, (required), commit message` `:param content: string, (required), the actual data in the file` `:param branch: string, (optional), branch to create the commit on. Defaults to the default branch of the repository` `:param committer: InputGitAuthor, (optional), if no information is given the authenticated user's information will be used. You must specify both a name and email.` `:param author: InputGitAuthor, (optional), if omitted this will be filled in with committer information. If passed, you must specify both a name and email.` `:rtype: {`

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

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

**delete\_file** (*path, message, sha, branch=NotSet, committer=NotSet, author=NotSet*)

This method delete a file in a repository :calls: `DELETE /repos/:owner/:repo/contents/:path` :param path: string, Required. The content path. :param message: string, Required. The commit message. :param sha: string, Required. The blob SHA of the file being replaced. :param branch: string. The branch name. Default: the repository's default branch (usually master) :param committer: InputGitAuthor, (optional), if no information is given the authenticated user's information will be used. You must specify both a name and email. :param author: InputGitAuthor, (optional), if omitted this will be filled in with committer information. If passed, you must specify both a name and email. :rtype: {

```
    '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\_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 https://developer.github.com/v3/repos/releases/#get-a-single-release`

**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\_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

**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

## 2.4.36 Rate

**class** `github.Rate.Rate`

This class represents Rates. The reference can be found here [http://developer.github.com/v3/rate\\_limit](http://developer.github.com/v3/rate_limit)

**limit**

**Type** integer

**remaining**

**Type** integer

**reset**

**Type** `datetime.datetime`

## 2.4.37 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

**author**  
**Type** *github.NamedUser.NamedUser*

**total**  
**Type** int

**weeks**  
**Type** list of *Week*

### 2.4.38 InstallationAuthorization

**class** `github.InstallationAuthorization.InstallationAuthorization`  
 This class represents InstallationAuthorizations

**token**  
**Type** string

**expires\_at**  
**Type** datetime

**on\_behalf\_of**  
**Type** *github.NamedUser.NamedUser*

### 2.4.39 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

## 2.4.40 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`

## 2.4.41 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 <<https://developer.github.com/v3/reactions/#list-reactions-for-a-commit-comment>>

**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`

## 2.4.42 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`

## 2.4.43 Plan

**class** `github.Plan.Plan`

This class represents Plans

**collaborators**

**Type** `integer`

**name**

**Type** `string`

**private\_repos**

**Type** `integer`

**space**

**Type** integer

#### 2.4.44 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

#### 2.4.45 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

#### 2.4.46 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

### 2.4.47 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`

## 2.4.48 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`

## 2.4.49 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

## 2.4.50 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

### 2.4.51 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

## 2.4.52 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

## 2.4.53 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

## 2.4.54 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`  
<<https://developer.github.com/v3/reactions/#list-reactions-for-an-issue-comment>>

**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`

## 2.4.55 SourceImport

**class** `github.SourceImport.SourceImport`

This class represents SourceImports. The reference can be found here [https://developer.github.com/v3/migration/source\\_imports/](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

## 2.4.56 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`

## 2.4.57 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:

## 2.4.58 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

### 2.4.59 RateLimit

**class** `github.RateLimit.RateLimit`

This class represents RateLimits. The reference can be found here [http://developer.github.com/v3/rate\\_limit](http://developer.github.com/v3/rate_limit)

**rate**  
 Type *class:github.Rate.Rate*

### 2.4.60 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

### 2.4.61 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

**draft**  
    Type bool

**prerelease**  
    Type bool

**author**  
    Type `github.GitAuthor.GitAuthor`

**created\_at**  
    Type `datetime.datetime`

**published\_at**  
    Type `datetime.datetime`

**url**  
    Type string

**upload\_url**  
    Type string

**html\_url**

**Type** string  
**tarball\_url**  
**Type** string  
**zipball\_url**  
**Type** string

## 2.4.62 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

## 2.4.63 Permissions

**class** `github.Permissions.Permissions`

This class represents Permissions

**admin**  
**Type** bool  
**pull**  
**Type** bool  
**push**

Type bool

## 2.4.64 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

**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*

**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*

**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** ()

**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\_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\_reviewer\_requests()**

**Calls** GET /repos/:owner/:repo/pulls/:number/requested\_reviewers

**Return type** `github.PaginatedList.PaginatedList` of `github.InspectionReviewers.InspectionReviewers`

**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`

## 2.4.65 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

## 2.4.66 Authorization

**class** `github.Authorization.Authorization`

This class represents Authorizations. The reference can be found here [https://developer.github.com/v3/oauth\\_authorized/](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

## 2.4.67 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

## 2.4.68 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

## 2.4.69 GitObject

**class** github.GitObject.**GitObject**

This class represents GitObjects

**sha**

Type string

**type**

Type string

**url**

Type string

## 2.4.70 CommitStats

**class** github.CommitStats.**CommitStats**

This class represents CommitStatses.

**additions**

Type integer

**deletions**

Type integer

**total**

Type integer

### 3.1 Stable versions

#### 3.1.1 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)

#### 3.1.2 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)

### 3.1.3 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)

### 3.1.4 Version 1.35 (July 10, 2017)

- Add Support for repository collaborator invitations.

### 3.1.5 Version 1.34 (abril 04, 2017)

- Add Support for Pull Request Reviews feature.

### 3.1.6 Version 1.32 (February 1, 2017)

- Support for Integrations installation endpoint (656e70e1)

### 3.1.7 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)

### 3.1.8 Version 1.30 (January 30, 2017)

- adds GitHub integrations (d60943d)

### 3.1.9 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)

### 3.1.10 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)

### 3.1.11 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)

### 3.1.12 Version 1.27.0 (August 12, 2016)

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

### 3.1.13 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()`

### 3.1.14 Version 1.25.2 (October 7th, 2014)

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

### 3.1.15 Version 1.25.1 (September 28th, 2014)

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

### 3.1.16 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

### 3.1.17 Version 1.24.1 (March 16th, 2014)

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

### 3.1.18 Version 1.24.0 (March 2nd, 2014)

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

### 3.1.19 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](#)



### 3.1.20 Version 1.22.0 (December 15th, 2013)

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

### 3.1.21 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

### 3.1.22 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

### 3.1.23 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)

### 3.1.24 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

### 3.1.25 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](#)

### 3.1.26 Version 1.16.0 (May 31th, 2013) (Concarneau edition)

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

### 3.1.27 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

### 3.1.28 Version 1.14.2 (April 25th, 2013)

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

### 3.1.29 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.

### 3.1.30 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

### 3.1.31 Version 1.13.1 (March 28nd, 2013)

- Fix login/password authentication for Python 3. Thank you sebastianstigler for reporting

### 3.1.32 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 pgoim
- Fix automated tests on Python 3.3. Thank you bkabrda for reporting

### 3.1.33 Version 1.12.2 (March 3rd, 2013)

- Fix major issue with Python 3: Json decoding was broken. Thank you bilderbuchi for reporting

### 3.1.34 Version 1.12.1 (February 20th, 2013)

- Nothing, but packaging/upload of 1.12.0 failed

### 3.1.35 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

### 3.1.36 Version 1.11.1 (February 9th, 2013) (London edition)

- Fix bug in lazy completion. Thank you ianozsvald for pinpointing it

### 3.1.37 Version 1.11.0 (February 7th, 2013)

- Fix bug in PaginatedList without url parameters. Thank you llimllib 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

### 3.1.38 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()`)

### 3.1.39 Version 1.9.1 (November 20th, 2012)

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

### 3.1.40 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

### 3.1.41 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](#)

### 3.1.42 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

### 3.1.43 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)

### 3.1.44 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

### 3.1.45 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

### 3.1.46 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

### 3.1.47 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

### 3.1.48 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

### 3.1.49 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

### 3.1.50 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, 3

github.GithubException, 20



## A

- a (github.StatsContributor.StatsContributor.Week attribute), 78
- accept\_invitation() (github.AuthenticatedUser.AuthenticatedUser method), 31
- accesskeyid (github.Download.Download attribute), 47
- acl (github.Download.Download attribute), 47
- active (github.Hook.Hook attribute), 40
- actor (github.Event.Event attribute), 53
- actor (github.IssueEvent.IssueEvent attribute), 42
- actual\_value (github.GithubException.BadAttributeException attribute), 21
- add\_membership() (github.Team.Team method), 38
- add\_to\_assignees() (github.Issue.Issue method), 36
- add\_to\_collaborators() (github.Repository.Repository method), 65
- add\_to\_emails() (github.AuthenticatedUser.AuthenticatedUser method), 25
- add\_to\_following() (github.AuthenticatedUser.AuthenticatedUser method), 25
- add\_to\_labels() (github.Issue.Issue method), 36
- add\_to\_members() (github.Team.Team method), 38
- add\_to\_public\_members() (github.Organization.Organization method), 57
- add\_to\_repos() (github.Team.Team method), 38
- add\_to\_starred() (github.AuthenticatedUser.AuthenticatedUser method), 25
- add\_to\_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 25
- add\_to\_watched() (github.AuthenticatedUser.AuthenticatedUser method), 25
- additions (github.CommitStats.CommitStats attribute), 102
- additions (github.File.File attribute), 86
- additions (github.PullRequest.PullRequest attribute), 96
- additions (github.StatsCodeFrequency.StatsCodeFrequency attribute), 83
- admin (github.Permissions.Permissions attribute), 95
- ahead\_by (github.Comparison.Comparison attribute), 89
- all (github.StatsParticipation.StatsParticipation attribute), 84
- app (github.Authorization.Authorization attribute), 100
- archive\_url (github.Repository.Repository attribute), 61
- archived (github.Repository.Repository attribute), 61
- assignee (github.Issue.Issue attribute), 34
- assignee (github.PullRequest.PullRequest attribute), 96
- assignees (github.Issue.Issue attribute), 34
- assignees (github.PullRequest.PullRequest attribute), 96
- assignees\_url (github.Repository.Repository attribute), 61
- AuthenticatedUser (class in github.AuthenticatedUser), 23
- author (github.Commit.Commit attribute), 33
- author (github.GitCommit.GitCommit attribute), 88
- author (github.GitRelease.GitRelease attribute), 94
- author (github.StatsContributor.StatsContributor attribute), 79
- Authorization (class in github.Authorization), 100
- AuthorizationApplication (class in github.AuthorizationApplication), 40
- authors\_count (github.SourceImport.SourceImport attribute), 91
- authors\_url (github.SourceImport.SourceImport attribute), 91
- avatar\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 23
- avatar\_url (github.NamedUser.NamedUser attribute), 49
- avatar\_url (github.Organization.Organization attribute), 55

## B

- BadAttributeException, 21
- BadCredentialsException, 21
- BadUserAgentException, 21
- base (github.PullRequest.PullRequest attribute), 96
- base\_commit (github.Comparison.Comparison attribute), 89
- behind\_by (github.Comparison.Comparison attribute), 89
- billing\_email (github.Organization.Organization attribute), 55

- bio (github.AuthenticatedUser.AuthenticatedUser attribute), 23
  - bio (github.NamedUser.NamedUser attribute), 49
  - blob\_url (github.File.File attribute), 86
  - blobs\_url (github.Repository.Repository attribute), 61
  - blog (github.AuthenticatedUser.AuthenticatedUser attribute), 23
  - blog (github.NamedUser.NamedUser attribute), 49
  - blog (github.Organization.Organization attribute), 55
  - body (github.CommitComment.CommitComment attribute), 82
  - body (github.GistComment.GistComment attribute), 79
  - body (github.GitRelease.GitRelease attribute), 94
  - body (github.Issue.Issue attribute), 34
  - body (github.IssueComment.IssueComment attribute), 90
  - body (github.PullRequest.PullRequest attribute), 96
  - body (github.PullRequestComment.PullRequestComment attribute), 31
  - body (github.PullRequestReview.PullRequestReview attribute), 93
  - body (github.StatusMessage.StatusMessage attribute), 89
  - Branch (class in github.Branch), 41
  - branches\_url (github.Repository.Repository attribute), 61
  - browser\_download\_url (github.GitReleaseAsset.GitReleaseAsset attribute), 85
  - bucket (github.Download.Download attribute), 47
- C**
- c (github.StatsContributor.StatsContributor.Week attribute), 78
  - change\_status (github.GistHistoryState.GistHistoryState attribute), 87
  - changed\_files (github.PullRequest.PullRequest attribute), 96
  - changes (github.File.File attribute), 86
  - clone\_url (github.Repository.Repository attribute), 62
  - closed\_at (github.Issue.Issue attribute), 35
  - closed\_at (github.PullRequest.PullRequest attribute), 96
  - closed\_by (github.Issue.Issue attribute), 35
  - closed\_issues (github.Milestone.Milestone attribute), 45
  - code (github.HookResponse.HookResponse attribute), 54
  - collaborators (github.AuthenticatedUser.AuthenticatedUser attribute), 23
  - collaborators (github.NamedUser.NamedUser attribute), 49
  - collaborators (github.Organization.Organization attribute), 55
  - collaborators (github.Plan.Plan attribute), 83
  - collaborators\_url (github.Repository.Repository attribute), 62
  - color (github.Label.Label attribute), 44
  - comments (github.Gist.Gist attribute), 80
  - comments (github.GistHistoryState.GistHistoryState attribute), 87
  - comments (github.Issue.Issue attribute), 35
  - comments (github.PullRequest.PullRequest attribute), 96
  - comments\_url (github.Commit.Commit attribute), 33
  - comments\_url (github.Gist.Gist attribute), 80
  - comments\_url (github.GistHistoryState.GistHistoryState attribute), 87
  - comments\_url (github.Issue.Issue attribute), 35
  - comments\_url (github.PullRequest.PullRequest attribute), 96
  - comments\_url (github.Repository.Repository attribute), 62
  - Commit (class in github.Commit), 33
  - commit (github.Branch.Branch attribute), 41
  - commit (github.Commit.Commit attribute), 33
  - commit (github.Tag.Tag attribute), 40
  - commit\_id (github.CommitComment.CommitComment attribute), 82
  - commit\_id (github.IssueEvent.IssueEvent attribute), 42
  - commit\_id (github.PullRequestComment.PullRequestComment attribute), 31
  - commit\_id (github.PullRequestReview.PullRequestReview attribute), 93
  - commit\_url (github.CommitCombinedStatus.CommitCombinedStatus attribute), 85
  - CommitCombinedStatus (class in github.CommitCombinedStatus), 85
  - CommitComment (class in github.CommitComment), 82
  - commits (github.Comparison.Comparison attribute), 89
  - commits (github.PullRequest.PullRequest attribute), 96
  - commits\_url (github.Gist.Gist attribute), 80
  - commits\_url (github.GistHistoryState.GistHistoryState attribute), 87
  - commits\_url (github.PullRequest.PullRequest attribute), 96
  - commits\_url (github.Repository.Repository attribute), 62
  - CommitStats (class in github.CommitStats), 102
  - CommitStatus (class in github.CommitStatus), 42
  - committed\_at (github.GistHistoryState.GistHistoryState attribute), 87
  - committer (github.Commit.Commit attribute), 33
  - committer (github.GitCommit.GitCommit attribute), 88
  - company (github.AuthenticatedUser.AuthenticatedUser attribute), 23
  - company (github.NamedUser.NamedUser attribute), 49
  - company (github.Organization.Organization attribute), 55
  - compare() (github.Repository.Repository method), 65
  - compare\_url (github.Repository.Repository attribute), 62
  - Comparison (class in github.Comparison), 89
  - config (github.Hook.Hook attribute), 40
  - content (github.ContentFile.ContentFile attribute), 32
  - content (github.GistFile.GistFile attribute), 84
  - content (github.GitBlob.GitBlob attribute), 60
  - content (github.Reaction.Reaction attribute), 54
  - content\_type (github.Download.Download attribute), 47

- content\_type (github.GitReleaseAsset.GitReleaseAsset attribute), 85
- ContentFile (class in github.ContentFile), 32
- contents\_url (github.File.File attribute), 86
- contents\_url (github.Repository.Repository attribute), 62
- context (github.CommitStatus.CommitStatus attribute), 42
- contexts (github.Branch.Branch attribute), 42
- contributions (github.NamedUser.NamedUser attribute), 49
- contributors\_url (github.Repository.Repository attribute), 62
- create\_authorization() (github.AuthenticatedUser.AuthenticatedUser method), 25
- create\_comment() (github.Commit.Commit method), 34
- create\_comment() (github.Gist.Gist method), 81
- create\_comment() (github.Issue.Issue method), 36
- create\_comment() (github.PullRequest.PullRequest method), 98
- create\_file() (github.Repository.Repository method), 70
- create\_fork() (github.AuthenticatedUser.AuthenticatedUser method), 26
- create\_fork() (github.Gist.Gist method), 81
- create\_fork() (github.Organization.Organization method), 57
- create\_from\_raw\_data() (github.MainClass.Github method), 6
- create\_gist() (github.AuthenticatedUser.AuthenticatedUser method), 26
- create\_git\_blob() (github.Repository.Repository method), 66
- create\_git\_commit() (github.Repository.Repository method), 66
- create\_git\_ref() (github.Repository.Repository method), 66
- create\_git\_tag() (github.Repository.Repository method), 66
- create\_git\_tree() (github.Repository.Repository method), 66
- create\_hook() (github.Organization.Organization method), 57
- create\_hook() (github.Repository.Repository method), 67
- create\_issue() (github.Repository.Repository method), 67
- create\_issue\_comment() (github.PullRequest.PullRequest method), 98
- create\_key() (github.AuthenticatedUser.AuthenticatedUser method), 26
- create\_key() (github.Repository.Repository method), 67
- create\_label() (github.Repository.Repository method), 67
- create\_milestone() (github.Repository.Repository method), 68
- create\_pull() (github.Repository.Repository method), 68
- create\_reaction() (github.CommitComment.CommitComment method), 83
- create\_reaction() (github.Issue.Issue method), 37
- create\_reaction() (github.IssueComment.IssueComment method), 90
- create\_reaction() (github.PullRequestComment.PullRequestComment method), 32
- create\_repo() (github.AuthenticatedUser.AuthenticatedUser method), 26
- create\_repo() (github.Organization.Organization method), 57
- create\_review() (github.PullRequest.PullRequest method), 98
- create\_review\_comment() (github.PullRequest.PullRequest method), 98
- create\_source\_import() (github.Repository.Repository method), 68
- create\_status() (github.Commit.Commit method), 34
- create\_team() (github.Organization.Organization method), 58
- created\_at (github.AuthenticatedUser.AuthenticatedUser attribute), 23
- created\_at (github.Authorization.Authorization attribute), 100
- created\_at (github.CommitComment.CommitComment attribute), 82
- created\_at (github.CommitStatus.CommitStatus attribute), 42
- created\_at (github.Download.Download attribute), 47
- created\_at (github.Event.Event attribute), 53
- created\_at (github.Gist.Gist attribute), 80
- created\_at (github.GistComment.GistComment attribute), 79
- created\_at (github.GistHistoryState.GistHistoryState attribute), 87
- created\_at (github.GitRelease.GitRelease attribute), 94
- created\_at (github.GitReleaseAsset.GitReleaseAsset attribute), 85
- created\_at (github.Hook.Hook attribute), 40
- created\_at (github.Invitation.Invitation attribute), 44
- created\_at (github.Issue.Issue attribute), 35
- created\_at (github.IssueComment.IssueComment attribute), 90
- created\_at (github.IssueEvent.IssueEvent attribute), 43
- created\_at (github.Milestone.Milestone attribute), 45
- created\_at (github.NamedUser.NamedUser attribute), 49
- created\_at (github.Organization.Organization attribute), 55
- created\_at (github.PullRequest.PullRequest attribute), 96
- created\_at (github.PullRequestComment.PullRequestComment attribute), 31
- created\_at (github.Reaction.Reaction attribute), 54
- created\_at (github.Repository.Repository attribute), 62
- created\_at (github.RepositoryKey.RepositoryKey attribute), 93

- created\_on (github.StatusMessage.StatusMessage attribute), 90
- creator (github.CommitStatus.CommitStatus attribute), 42
- creator (github.Milestone.Milestone attribute), 45
- ## D
- d (github.StatsContributor.StatsContributor.Week attribute), 78
- data (github.GithubException.GithubException attribute), 20
- date (github.GitAuthor.GitAuthor attribute), 61
- days (github.StatsCommitActivity.StatsCommitActivity attribute), 102
- default\_branch (github.Repository.Repository attribute), 62
- delete() (github.Authorization.Authorization method), 101
- delete() (github.CommitComment.CommitComment method), 82
- delete() (github.Download.Download method), 48
- delete() (github.Gist.Gist method), 81
- delete() (github.GistComment.GistComment method), 79
- delete() (github.GitRef.GitRef method), 92
- delete() (github.Hook.Hook method), 41
- delete() (github.IssueComment.IssueComment method), 90
- delete() (github.Label.Label method), 45
- delete() (github.Milestone.Milestone method), 46
- delete() (github.PullRequestComment.PullRequestComment method), 32
- delete() (github.Reaction.Reaction method), 54
- delete() (github.Repository.Repository method), 68
- delete() (github.RepositoryKey.RepositoryKey method), 94
- delete() (github.Team.Team method), 39
- delete() (github.UserKey.UserKey method), 43
- delete\_asset() (github.GitReleaseAsset.GitReleaseAsset method), 85
- delete\_file() (github.Repository.Repository method), 71
- delete\_hook() (github.Organization.Organization method), 58
- delete\_labels() (github.Issue.Issue method), 36
- deletions (github.CommitStats.CommitStats attribute), 102
- deletions (github.File.File attribute), 86
- deletions (github.PullRequest.PullRequest attribute), 96
- deletions (github.StatsCodeFrequency.StatsCodeFrequency attribute), 83
- description (github.CommitStatus.CommitStatus attribute), 42
- description (github.Download.Download attribute), 47
- description (github.Gist.Gist attribute), 80
- description (github.GistHistoryState.GistHistoryState attribute), 87
- description (github.Milestone.Milestone attribute), 46
- description (github.Repository.Repository attribute), 62
- diff\_hunk (github.PullRequestComment.PullRequestComment attribute), 31
- diff\_url (github.Comparison.Comparison attribute), 89
- diff\_url (github.IssuePullRequest.IssuePullRequest attribute), 55
- diff\_url (github.PullRequest.PullRequest attribute), 96
- disk\_usage (github.AuthenticatedUser.AuthenticatedUser attribute), 23
- disk\_usage (github.NamedUser.NamedUser attribute), 49
- disk\_usage (github.Organization.Organization attribute), 55
- Download (class in github.Download), 47
- download\_count (github.Download.Download attribute), 47
- download\_count (github.GitReleaseAsset.GitReleaseAsset attribute), 85
- downloads\_url (github.Repository.Repository attribute), 62
- draft (github.GitRelease.GitRelease attribute), 94
- due\_on (github.Milestone.Milestone attribute), 46
- dump() (github.MainClass.Github method), 7
- ## E
- edit() (github.AuthenticatedUser.AuthenticatedUser method), 27
- edit() (github.Authorization.Authorization method), 101
- edit() (github.CommitComment.CommitComment method), 83
- edit() (github.Gist.Gist method), 81
- edit() (github.GistComment.GistComment method), 80
- edit() (github.GitRef.GitRef method), 92
- edit() (github.Hook.Hook method), 41
- edit() (github.Issue.Issue method), 36
- edit() (github.IssueComment.IssueComment method), 90
- edit() (github.Label.Label method), 45
- edit() (github.Milestone.Milestone method), 46
- edit() (github.Organization.Organization method), 58
- edit() (github.PullRequest.PullRequest method), 98
- edit() (github.PullRequestComment.PullRequestComment method), 32
- edit() (github.Repository.Repository method), 68
- edit() (github.Team.Team method), 39
- edit() (github.UserKey.UserKey method), 43
- edit\_hook() (github.Organization.Organization method), 58
- email (github.AuthenticatedUser.AuthenticatedUser attribute), 23
- email (github.GitAuthor.GitAuthor attribute), 61
- email (github.NamedUser.NamedUser attribute), 49
- email (github.Organization.Organization attribute), 55

- enable\_console\_debug\_logging() (in module github), 20  
 encoding (github.ContentFile.ContentFile attribute), 32  
 encoding (github.GitBlob.GitBlob attribute), 60  
 enforcement\_level (github.Branch.Branch attribute), 42  
 etag (github.GithubObject.GithubObject attribute), 22  
 Event (class in github.Event), 53  
 event (github.IssueEvent.IssueEvent attribute), 43  
 events (github.Hook.Hook attribute), 40  
 events (github.HookDescription.HookDescription attribute), 44  
 events\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 23  
 events\_url (github.Issue.Issue attribute), 35  
 events\_url (github.NamedUser.NamedUser attribute), 49  
 events\_url (github.Organization.Organization attribute), 55  
 events\_url (github.Repository.Repository attribute), 62  
 expected\_type (github.GithubException.BadAttributeException attribute), 21  
 expirationdate (github.Download.Download attribute), 47  
 expires\_at (github.InstallationAuthorization.InstallationAuthorization attribute), 79
- ## F
- File (class in github.File), 86  
 filename (github.File.File attribute), 86  
 filename (github.GistFile.GistFile attribute), 84  
 files (github.Commit.Commit attribute), 33  
 files (github.Comparison.Comparison attribute), 89  
 files (github.Gist.Gist attribute), 80  
 files (github.GistHistoryState.GistHistoryState attribute), 87  
 FIX\_REPO\_GET\_GIT\_REF (github.MainClass.Github attribute), 3  
 followers (github.AuthenticatedUser.AuthenticatedUser attribute), 23  
 followers (github.NamedUser.NamedUser attribute), 49  
 followers (github.Organization.Organization attribute), 56  
 followers\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 23  
 followers\_url (github.NamedUser.NamedUser attribute), 49  
 following (github.AuthenticatedUser.AuthenticatedUser attribute), 23  
 following (github.NamedUser.NamedUser attribute), 49  
 following (github.Organization.Organization attribute), 56  
 following\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 23  
 following\_url (github.NamedUser.NamedUser attribute), 50  
 fork (github.Repository.Repository attribute), 62  
 fork\_of (github.Gist.Gist attribute), 80  
 forks (github.Gist.Gist attribute), 80  
 forks (github.GistHistoryState.GistHistoryState attribute), 87  
 forks (github.Repository.Repository attribute), 62  
 forks\_count (github.Repository.Repository attribute), 62  
 forks\_url (github.Gist.Gist attribute), 80  
 forks\_url (github.GistHistoryState.GistHistoryState attribute), 87  
 forks\_url (github.Repository.Repository attribute), 62  
 full\_name (github.Repository.Repository attribute), 62
- ## G
- get() (github.StatsPunchCard.StatsPunchCard method), 47  
 get\_repr\_\_() (github.GithubObject.GithubObject method), 22  
 get\_api\_status() (github.MainClass.Github method), 7  
 get\_api\_status\_messages() (github.MainClass.Github method), 7  
 get\_archive\_link() (github.Repository.Repository method), 69  
 get\_assignees() (github.Repository.Repository method), 69  
 get\_authorization() (github.AuthenticatedUser.AuthenticatedUser method), 27  
 get\_authorizations() (github.AuthenticatedUser.AuthenticatedUser method), 27  
 get\_branch() (github.Repository.Repository method), 69  
 get\_branches() (github.Repository.Repository method), 69  
 get\_collaborators() (github.Repository.Repository method), 69  
 get\_combined\_status() (github.Commit.Commit method), 34  
 get\_comment() (github.Gist.Gist method), 81  
 get\_comment() (github.Issue.Issue method), 36  
 get\_comment() (github.PullRequest.PullRequest method), 99  
 get\_comment() (github.Repository.Repository method), 69  
 get\_comments() (github.Commit.Commit method), 34  
 get\_comments() (github.Gist.Gist method), 81  
 get\_comments() (github.Issue.Issue method), 36  
 get\_comments() (github.PullRequest.PullRequest method), 99  
 get\_comments() (github.Repository.Repository method), 69  
 get\_commit() (github.Repository.Repository method), 70  
 get\_commits() (github.PullRequest.PullRequest method), 99  
 get\_commits() (github.Repository.Repository method), 70  
 get\_contents() (github.Repository.Repository method), 70  
 get\_contributors() (github.Repository.Repository method), 71

- [get\\_dir\\_contents\(\)](#) (github.Repository.Repository method), 71  
[get\\_download\(\)](#) (github.Repository.Repository method), 71  
[get\\_downloads\(\)](#) (github.Repository.Repository method), 71  
[get\\_emails\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 27  
[get\\_emojis\(\)](#) (github.MainClass.Github method), 6  
[get\\_events\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 27  
[get\\_events\(\)](#) (github.Issue.Issue method), 37  
[get\\_events\(\)](#) (github.NamedUser.NamedUser method), 51  
[get\\_events\(\)](#) (github.Organization.Organization method), 59  
[get\\_events\(\)](#) (github.Repository.Repository method), 71  
[get\\_file\\_contents\(\)](#) (github.Repository.Repository method), 70  
[get\\_files\(\)](#) (github.PullRequest.PullRequest method), 99  
[get\\_followers\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 27  
[get\\_followers\(\)](#) (github.NamedUser.NamedUser method), 51  
[get\\_following\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 27  
[get\\_following\(\)](#) (github.NamedUser.NamedUser method), 51  
[get\\_forks\(\)](#) (github.Repository.Repository method), 71  
[get\\_gist\(\)](#) (github.MainClass.Github method), 4  
[get\\_gists\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 27  
[get\\_gists\(\)](#) (github.MainClass.Github method), 5  
[get\\_gists\(\)](#) (github.NamedUser.NamedUser method), 51  
[get\\_git\\_blob\(\)](#) (github.Repository.Repository method), 72  
[get\\_git\\_commit\(\)](#) (github.Repository.Repository method), 72  
[get\\_git\\_ref\(\)](#) (github.Repository.Repository method), 72  
[get\\_git\\_refs\(\)](#) (github.Repository.Repository method), 72  
[get\\_git\\_tag\(\)](#) (github.Repository.Repository method), 72  
[get\\_git\\_tree\(\)](#) (github.Repository.Repository method), 72  
[get\\_gitignore\\_template\(\)](#) (github.MainClass.Github method), 6  
[get\\_gitignore\\_templates\(\)](#) (github.MainClass.Github method), 6  
[get\\_hook\(\)](#) (github.MainClass.Github method), 6  
[get\\_hook\(\)](#) (github.Organization.Organization method), 59  
[get\\_hook\(\)](#) (github.Repository.Repository method), 72  
[get\\_hooks\(\)](#) (github.MainClass.Github method), 6  
[get\\_hooks\(\)](#) (github.Organization.Organization method), 59  
[get\\_hooks\(\)](#) (github.Repository.Repository method), 72  
[get\\_installation\(\)](#) (github.MainClass.Github method), 7  
[get\\_issue\(\)](#) (github.Repository.Repository method), 72  
[get\\_issue\\_comment\(\)](#) (github.PullRequest.PullRequest method), 99  
[get\\_issue\\_comments\(\)](#) (github.PullRequest.PullRequest method), 99  
[get\\_issues\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 28  
[get\\_issues\(\)](#) (github.Organization.Organization method), 59  
[get\\_issues\(\)](#) (github.Repository.Repository method), 73  
[get\\_issues\\_comments\(\)](#) (github.Repository.Repository method), 73  
[get\\_issues\\_event\(\)](#) (github.Repository.Repository method), 73  
[get\\_issues\\_events\(\)](#) (github.Repository.Repository method), 73  
[get\\_key\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 28  
[get\\_key\(\)](#) (github.Repository.Repository method), 73  
[get\\_keys\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 28  
[get\\_keys\(\)](#) (github.NamedUser.NamedUser method), 51  
[get\\_keys\(\)](#) (github.Repository.Repository method), 73  
[get\\_label\(\)](#) (github.Repository.Repository method), 74  
[get\\_labels\(\)](#) (github.Issue.Issue method), 37  
[get\\_labels\(\)](#) (github.Milestone.Milestone method), 46  
[get\\_labels\(\)](#) (github.Repository.Repository method), 74  
[get\\_languages\(\)](#) (github.Repository.Repository method), 74  
[get\\_last\\_api\\_status\\_message\(\)](#) (github.MainClass.Github method), 7  
[get\\_latest\\_release\(\)](#) (github.Repository.Repository method), 76  
[get\\_members\(\)](#) (github.Organization.Organization method), 59  
[get\\_members\(\)](#) (github.Team.Team method), 39  
[get\\_milestone\(\)](#) (github.Repository.Repository method), 74  
[get\\_milestones\(\)](#) (github.Repository.Repository method), 74  
[get\\_network\\_events\(\)](#) (github.Repository.Repository method), 74  
[get\\_notification\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 28  
[get\\_notifications\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 28  
[get\\_organization\(\)](#) (github.MainClass.Github method), 4  
[get\\_organization\\_events\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_orgs\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_orgs\(\)](#) (github.NamedUser.NamedUser method), 51  
[get\\_protected\\_branch\(\)](#) (github.Repository.Repository method), 69



- [get\\_public\\_events\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_public\\_members\(\)](#) (github.Organization.Organization method), 59  
[get\\_public\\_received\\_events\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_pull\(\)](#) (github.Repository.Repository method), 74  
[get\\_pulls\(\)](#) (github.Repository.Repository method), 74  
[get\\_pulls\\_comments\(\)](#) (github.Repository.Repository method), 75  
[get\\_pulls\\_review\\_comments\(\)](#) (github.Repository.Repository method), 75  
[get\\_rate\\_limit\(\)](#) (github.MainClass.Github method), 4  
[get\\_reactions\(\)](#) (github.CommitComment.CommitComment method), 83  
[get\\_reactions\(\)](#) (github.Issue.Issue method), 37  
[get\\_reactions\(\)](#) (github.IssueComment.IssueComment method), 90  
[get\\_reactions\(\)](#) (github.PullRequestComment.PullRequestComment method), 32  
[get\\_readme\(\)](#) (github.Repository.Repository method), 75  
[get\\_received\\_events\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_release\(\)](#) (github.Repository.Repository method), 76  
[get\\_releases\(\)](#) (github.Repository.Repository method), 76  
[get\\_repo\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_repo\(\)](#) (github.MainClass.Github method), 4  
[get\\_repo\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_repo\(\)](#) (github.Organization.Organization method), 59  
[get\\_repos\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_repos\(\)](#) (github.Installation.Installation method), 53  
[get\\_repos\(\)](#) (github.MainClass.Github method), 4  
[get\\_repos\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_repos\(\)](#) (github.Organization.Organization method), 59  
[get\\_repos\(\)](#) (github.Team.Team method), 39  
[get\\_review\(\)](#) (github.PullRequest.PullRequest method), 99  
[get\\_review\\_comment\(\)](#) (github.PullRequest.PullRequest method), 99  
[get\\_review\\_comments\(\)](#) (github.PullRequest.PullRequest method), 99  
[get\\_reviewer\\_requests\(\)](#) (github.PullRequest.PullRequest method), 100  
[get\\_reviews\(\)](#) (github.PullRequest.PullRequest method), 100  
[get\\_source\\_import\(\)](#) (github.Repository.Repository method), 75  
[get\\_stargazers\(\)](#) (github.Repository.Repository method), 75  
[get\\_stargazers\\_with\\_dates\(\)](#) (github.Repository.Repository method), 75  
[get\\_starred\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_starred\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_starred\\_gists\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_stats\\_code\\_frequency\(\)](#) (github.Repository.Repository method), 76  
[get\\_stats\\_commit\\_activity\(\)](#) (github.Repository.Repository method), 76  
[get\\_stats\\_contributors\(\)](#) (github.Repository.Repository method), 75  
[get\\_stats\\_participation\(\)](#) (github.Repository.Repository method), 76  
[get\\_stats\\_punch\\_card\(\)](#) (github.Repository.Repository method), 76  
[get\\_status\(\)](#) (github.GitRef.GitRef method), 92  
[get\\_statuses\(\)](#) (github.Commit.Commit method), 34  
[get\\_statuses\(\)](#) (github.GitRef.GitRef method), 92  
[get\\_subscribers\(\)](#) (github.Repository.Repository method), 76  
[get\\_subscriptions\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_subscriptions\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_tags\(\)](#) (github.Repository.Repository method), 76  
[get\\_team\(\)](#) (github.Organization.Organization method), 60  
[get\\_teams\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 29  
[get\\_teams\(\)](#) (github.Organization.Organization method), 60  
[get\\_teams\(\)](#) (github.Repository.Repository method), 76  
[get\\_user\(\)](#) (github.MainClass.Github method), 4  
[get\\_user\\_issues\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 28  
[get\\_users\(\)](#) (github.MainClass.Github method), 4  
[get\\_watched\(\)](#) (github.AuthenticatedUser.AuthenticatedUser method), 30  
[get\\_watched\(\)](#) (github.NamedUser.NamedUser method), 52  
[get\\_watchers\(\)](#) (github.Repository.Repository method), 76  
[Gist](#) (class in github.Gist), 80  
[GistComment](#) (class in github.GistComment), 79  
[GistFile](#) (class in github.GistFile), 84  
[GistHistoryState](#) (class in github.GistHistoryState), 87  
[gists\\_url](#) (github.AuthenticatedUser.AuthenticatedUser attribute), 23  
[gists\\_url](#) (github.NamedUser.NamedUser attribute), 50  
[git\\_commits\\_url](#) (github.Repository.Repository attribute), 62

- git\_pull\_url (github.Gist.Gist attribute), 80
  - git\_pull\_url (github.GistHistoryState.GistHistoryState attribute), 87
  - git\_push\_url (github.Gist.Gist attribute), 80
  - git\_push\_url (github.GistHistoryState.GistHistoryState attribute), 87
  - git\_refs\_url (github.Repository.Repository attribute), 63
  - git\_tags\_url (github.Repository.Repository attribute), 63
  - git\_url (github.ContentFile.ContentFile attribute), 32
  - git\_url (github.Repository.Repository attribute), 63
  - GitAuthor (class in github.GitAuthor), 61
  - GitBlob (class in github.GitBlob), 60
  - GitCommit (class in github.GitCommit), 88
  - Github (class in github.MainClass), 3
  - github (module), 3
  - github.GithubException (module), 20
  - GithubException, 20
  - GithubObject (class in github.GithubObject), 22
  - GitignoreTemplate (class in github.GitignoreTemplate), 53
  - GitObject (class in github.GitObject), 102
  - GitRef (class in github.GitRef), 92
  - GitRelease (class in github.GitRelease), 94
  - GitReleaseAsset (class in github.GitReleaseAsset), 84
  - GitTag (class in github.GitTag), 45
  - GitTree (class in github.GitTree), 101
  - GitTreeElement (class in github.GitTreeElement), 48
  - gravatar\_id (github.AuthenticatedUser.AuthenticatedUser attribute), 23
  - gravatar\_id (github.NamedUser.NamedUser attribute), 50
  - gravatar\_id (github.Organization.Organization attribute), 56
- ## H
- has\_downloads (github.Repository.Repository attribute), 63
  - has\_in\_assignees() (github.Repository.Repository method), 77
  - has\_in\_collaborators() (github.Repository.Repository method), 77
  - has\_in\_following() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - has\_in\_following() (github.NamedUser.NamedUser method), 52
  - has\_in\_members() (github.Organization.Organization method), 60
  - has\_in\_members() (github.Team.Team method), 39
  - has\_in\_public\_members() (github.Organization.Organization method), 60
  - has\_in\_repos() (github.Team.Team method), 39
  - has\_in\_starred() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - has\_in\_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - has\_in\_watched() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - has\_issues (github.Repository.Repository attribute), 63
  - has\_large\_files (github.SourceImport.SourceImport attribute), 91
  - has\_wiki (github.Repository.Repository attribute), 63
  - head (github.PullRequest.PullRequest attribute), 96
  - hireable (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - hireable (github.NamedUser.NamedUser attribute), 50
  - history (github.Gist.Gist attribute), 80
  - history (github.GistHistoryState.GistHistoryState attribute), 87
  - homepage (github.Repository.Repository attribute), 63
  - Hook (class in github.Hook), 40
  - HookDescription (class in github.HookDescription), 44
  - HookResponse (class in github.HookResponse), 54
  - hooks\_url (github.Repository.Repository attribute), 63
  - html\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - html\_url (github.Commit.Commit attribute), 33
  - html\_url (github.CommitComment.CommitComment attribute), 82
  - html\_url (github.Comparison.Comparison attribute), 89
  - html\_url (github.ContentFile.ContentFile attribute), 32
  - html\_url (github.Download.Download attribute), 47
  - html\_url (github.Gist.Gist attribute), 80
  - html\_url (github.GistHistoryState.GistHistoryState attribute), 87
  - html\_url (github.GitCommit.GitCommit attribute), 88
  - html\_url (github.GitRelease.GitRelease attribute), 94
  - html\_url (github.Invitation.Invitation attribute), 44
  - html\_url (github.Issue.Issue attribute), 35
  - html\_url (github.IssueComment.IssueComment attribute), 90
  - html\_url (github.IssuePullRequest.IssuePullRequest attribute), 55
  - html\_url (github.NamedUser.NamedUser attribute), 50
  - html\_url (github.Organization.Organization attribute), 56
  - html\_url (github.PullRequest.PullRequest attribute), 96
  - html\_url (github.PullRequestComment.PullRequestComment attribute), 32
  - html\_url (github.PullRequestReview.PullRequestReview attribute), 93
  - html\_url (github.Repository.Repository attribute), 63
  - html\_url (github.SourceImport.SourceImport attribute), 91
- ## I
- id (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - id (github.Authorization.Authorization attribute), 101
  - id (github.CommitComment.CommitComment attribute), 82

- id (github.CommitStatus.CommitStatus attribute), 42
  - id (github.Download.Download attribute), 47
  - id (github.Event.Event attribute), 53
  - id (github.Gist.Gist attribute), 80
  - id (github.GistComment.GistComment attribute), 79
  - id (github.GistHistoryState.GistHistoryState attribute), 87
  - id (github.GitRelease.GitRelease attribute), 94
  - id (github.GitReleaseAsset.GitReleaseAsset attribute), 84
  - id (github.Hook.Hook attribute), 40
  - id (github.Invitation.Invitation attribute), 44
  - id (github.Issue.Issue attribute), 35
  - id (github.IssueComment.IssueComment attribute), 90
  - id (github.IssueEvent.IssueEvent attribute), 43
  - id (github.Milestone.Milestone attribute), 46
  - id (github.NamedUser.NamedUser attribute), 50
  - id (github.Notification.Notification attribute), 95
  - id (github.Organization.Organization attribute), 56
  - id (github.PullRequest.PullRequest attribute), 97
  - id (github.PullRequestComment.PullRequestComment attribute), 31
  - id (github.PullRequestReview.PullRequestReview attribute), 92
  - id (github.PullRequestReviewerRequest.PullRequestReviewerRequest attribute), 49
  - id (github.Reaction.Reaction attribute), 54
  - id (github.Repository.Repository attribute), 63
  - id (github.RepositoryKey.RepositoryKey attribute), 93
  - id (github.Team.Team attribute), 37
  - id (github.UserKey.UserKey attribute), 43
  - InputFileContent (class in github.InputFileContent), 22
  - InputGitAuthor (class in github.InputGitAuthor), 22
  - InputGitTreeElement (class in github.InputGitTreeElement), 22
  - Installation (class in github.Installation), 53
  - InstallationAuthorization (class in github.InstallationAuthorization), 79
  - Invitation (class in github.Invitation), 44
  - is\_merged() (github.PullRequest.PullRequest method), 100
  - is\_starred() (github.Gist.Gist method), 81
  - Issue (class in github.Issue), 34
  - issue (github.IssueEvent.IssueEvent attribute), 43
  - issue\_comment\_url (github.Repository.Repository attribute), 63
  - issue\_events\_url (github.Repository.Repository attribute), 63
  - issue\_url (github.IssueComment.IssueComment attribute), 90
  - issue\_url (github.PullRequest.PullRequest attribute), 97
  - IssueComment (class in github.IssueComment), 90
  - IssueEvent (class in github.IssueEvent), 42
  - IssuePullRequest (class in github.IssuePullRequest), 55
  - issues\_url (github.Repository.Repository attribute), 63
- ## K
- key (github.RepositoryKey.RepositoryKey attribute), 93
  - key (github.UserKey.UserKey attribute), 43
  - keys\_url (github.Repository.Repository attribute), 63
- ## L
- Label (class in github.Label), 44
  - label (github.GitReleaseAsset.GitReleaseAsset attribute), 84
  - label (github.PullRequestPart.PullRequestPart attribute), 54
  - labels (github.Issue.Issue attribute), 35
  - labels\_url (github.Issue.Issue attribute), 35
  - labels\_url (github.Milestone.Milestone attribute), 46
  - labels\_url (github.Repository.Repository attribute), 63
  - language (github.GistFile.GistFile attribute), 84
  - language (github.Repository.Repository attribute), 63
  - languages\_url (github.Repository.Repository attribute), 63
  - large\_files\_count (github.SourceImport.SourceImport attribute), 91
  - large\_files\_size (github.SourceImport.SourceImport attribute), 91
  - last\_modified (github.GithubObject.GithubObject attribute), 22
  - last\_read\_at (github.Notification.Notification attribute), 95
  - last\_response (github.Hook.Hook attribute), 40
  - last\_updated (github.Status.Status attribute), 61
  - latest\_comment\_url (github.NotificationSubject.NotificationSubject attribute), 100
  - in legacy\_search\_issues() (github.Repository.Repository method), 77
  - limit (github.Rate.Rate attribute), 78
  - in line (github.CommitComment.CommitComment attribute), 82
  - load() (github.MainClass.Github method), 7
  - location (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - location (github.NamedUser.NamedUser attribute), 50
  - location (github.Organization.Organization attribute), 56
  - login (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - login (github.NamedUser.NamedUser attribute), 50
  - login (github.Organization.Organization attribute), 56
  - login (github.PullRequestReviewerRequest.PullRequestReviewerRequest attribute), 49
- ## M
- master\_branch (github.Repository.Repository attribute), 63
  - members\_count (github.Team.Team attribute), 37
  - members\_url (github.Organization.Organization attribute), 56

- members\_url (github.Team.Team attribute), 38
  - merge() (github.PullRequest.PullRequest method), 100
  - merge() (github.Repository.Repository method), 77
  - merge\_base\_commit (github.Comparison.Comparison attribute), 89
  - merge\_commit\_sha (github.PullRequest.PullRequest attribute), 97
  - mergeable (github.PullRequest.PullRequest attribute), 97
  - mergeable\_state (github.PullRequest.PullRequest attribute), 97
  - merged (github.PullRequest.PullRequest attribute), 97
  - merged (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 86
  - merged\_at (github.PullRequest.PullRequest attribute), 97
  - merged\_by (github.PullRequest.PullRequest attribute), 97
  - merges\_url (github.Repository.Repository attribute), 64
  - message (github.GitCommit.GitCommit attribute), 88
  - message (github.GitTag.GitTag attribute), 45
  - message (github.HookResponse.HookResponse attribute), 54
  - message (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 86
  - Milestone (class in github.Milestone), 45
  - milestone (github.Issue.Issue attribute), 35
  - milestone (github.PullRequest.PullRequest attribute), 97
  - milestones\_url (github.Repository.Repository attribute), 64
  - mime\_type (github.Download.Download attribute), 47
  - mirror\_url (github.Repository.Repository attribute), 64
  - mode (github.GitTreeElement.GitTreeElement attribute), 48
- ## N
- name (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - name (github.AuthorizationApplication.AuthorizationApplication attribute), 40
  - name (github.Branch.Branch attribute), 41
  - name (github.ContentFile.ContentFile attribute), 32
  - name (github.Download.Download attribute), 47
  - name (github.GitAuthor.GitAuthor attribute), 61
  - name (github.GitignoreTemplate.GitignoreTemplate attribute), 53
  - name (github.GitReleaseAsset.GitReleaseAsset attribute), 84
  - name (github.Hook.Hook attribute), 40
  - name (github.HookDescription.HookDescription attribute), 44
  - name (github.Label.Label attribute), 44
  - name (github.NamedUser.NamedUser attribute), 50
  - name (github.Organization.Organization attribute), 56
  - name (github.Plan.Plan attribute), 83
  - name (github.Repository.Repository attribute), 64
  - name (github.Tag.Tag attribute), 40
  - name (github.Team.Team attribute), 38
  - NamedUser (class in github.NamedUser), 49
  - network\_count (github.Repository.Repository attribute), 64
  - note (github.Authorization.Authorization attribute), 101
  - note\_url (github.Authorization.Authorization attribute), 101
  - Notification (class in github.Notification), 95
  - notifications\_url (github.Repository.Repository attribute), 64
  - NotificationSubject (class in github.NotificationSubject), 100
  - number (github.Issue.Issue attribute), 35
  - number (github.Milestone.Milestone attribute), 46
  - number (github.PullRequest.PullRequest attribute), 97
- ## O
- oauth\_scopes (github.MainClass.Github attribute), 4
  - object (github.GitRef.GitRef attribute), 92
  - object (github.GitTag.GitTag attribute), 45
  - on\_half\_of (github.InstallationAuthorization.InstallationAuthorization attribute), 79
  - open\_issues (github.Milestone.Milestone attribute), 46
  - open\_issues (github.Repository.Repository attribute), 64
  - open\_issues\_count (github.Repository.Repository attribute), 64
  - org (github.Event.Event attribute), 53
  - Organization (class in github.Organization), 55
  - organization (github.Repository.Repository attribute), 64
  - organization (github.Team.Team attribute), 38
  - organizations\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - organizations\_url (github.NamedUser.NamedUser attribute), 50
  - original\_commit\_id (github.PullRequestComment.PullRequestComment attribute), 31
  - original\_position (github.PullRequestComment.PullRequestComment attribute), 31
  - owned\_private\_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - owned\_private\_repos (github.NamedUser.NamedUser attribute), 50
  - owned\_private\_repos (github.Organization.Organization attribute), 56
  - owner (github.Gist.Gist attribute), 81
  - owner (github.GistHistoryState.GistHistoryState attribute), 88
  - owner (github.Repository.Repository attribute), 64
  - owner (github.StatsParticipation.StatsParticipation attribute), 84
- ## P
- PaginatedList (class in github.PaginatedList), 21
  - parent (github.Repository.Repository attribute), 64

- parents (github.Commit.Commit attribute), 33
  - parents (github.GitCommit.GitCommit attribute), 88
  - patch (github.File.File attribute), 86
  - patch\_url (github.Comparison.Comparison attribute), 89
  - patch\_url (github.IssuePullRequest.IssuePullRequest attribute), 55
  - patch\_url (github.PullRequest.PullRequest attribute), 97
  - path (github.CommitComment.CommitComment attribute), 82
  - path (github.ContentFile.ContentFile attribute), 33
  - path (github.Download.Download attribute), 47
  - path (github.GitTreeElement.GitTreeElement attribute), 48
  - path (github.PullRequestComment.PullRequestComment attribute), 31
  - payload (github.Event.Event attribute), 53
  - per\_page (github.MainClass.Github attribute), 3
  - permalink\_url (github.Comparison.Comparison attribute), 89
  - permission (github.Team.Team attribute), 38
  - Permissions (class in github.Permissions), 95
  - permissions (github.Invitation.Invitation attribute), 44
  - permissions (github.NamedUser.NamedUser attribute), 50
  - permissions (github.Repository.Repository attribute), 64
  - ping() (github.Hook.Hook method), 41
  - ping\_url (github.Hook.Hook attribute), 41
  - Plan (class in github.Plan), 83
  - plan (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - plan (github.NamedUser.NamedUser attribute), 50
  - plan (github.Organization.Organization attribute), 56
  - policy (github.Download.Download attribute), 47
  - position (github.CommitComment.CommitComment attribute), 82
  - position (github.PullRequestComment.PullRequestComment attribute), 31
  - prefix (github.Download.Download attribute), 48
  - prerelease (github.GitRelease.GitRelease attribute), 94
  - previous\_filename (github.File.File attribute), 86
  - private (github.Repository.Repository attribute), 64
  - private\_gists (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - private\_gists (github.NamedUser.NamedUser attribute), 50
  - private\_gists (github.Organization.Organization attribute), 56
  - private\_repos (github.Plan.Plan attribute), 83
  - protect\_branch() (github.Repository.Repository method), 77
  - protected (github.Branch.Branch attribute), 42
  - public (github.Event.Event attribute), 53
  - public (github.Gist.Gist attribute), 81
  - public (github.GistHistoryState.GistHistoryState attribute), 88
  - public\_gists (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - public\_gists (github.NamedUser.NamedUser attribute), 50
  - public\_gists (github.Organization.Organization attribute), 56
  - public\_members\_url (github.Organization.Organization attribute), 56
  - public\_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - public\_repos (github.NamedUser.NamedUser attribute), 50
  - public\_repos (github.Organization.Organization attribute), 56
  - published\_at (github.GitRelease.GitRelease attribute), 94
  - pull (github.Permissions.Permissions attribute), 95
  - pull\_request (github.Issue.Issue attribute), 35
  - pull\_request\_url (github.PullRequestComment.PullRequestComment attribute), 31
  - pull\_request\_url (github.PullRequestReview.PullRequestReview attribute), 93
  - PullRequest (class in github.PullRequest), 96
  - PullRequestComment (class in github.PullRequestComment), 31
  - PullRequestMergeStatus (class in github.PullRequestMergeStatus), 86
  - PullRequestPart (class in github.PullRequestPart), 54
  - PullRequestReview (class in github.PullRequestReview), 92
  - PullRequestReviewerRequest (class in github.PullRequestReviewerRequest), 49
  - pulls\_url (github.Repository.Repository attribute), 64
  - push (github.Permissions.Permissions attribute), 95
  - pushed\_at (github.Repository.Repository attribute), 64
- ## R
- Rate (class in github.Rate), 78
  - rate (github.RateLimit.RateLimit attribute), 93
  - rate\_limiting (github.MainClass.Github attribute), 4
  - rate\_limiting\_resetime (github.MainClass.Github attribute), 4
  - RateLimit (class in github.RateLimit), 93
  - RateLimitExceededException, 21
  - raw\_data (github.GithubObject.GithubObject attribute), 22
  - raw\_headers (github.GithubObject.GithubObject attribute), 22
  - raw\_url (github.File.File attribute), 86
  - raw\_url (github.GistFile.GistFile attribute), 84
  - Reaction (class in github.Reaction), 54
  - read\_only (github.RepositoryKey.RepositoryKey attribute), 94

- reason (github.Notification.Notification attribute), 95
  - received\_events\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - received\_events\_url (github.NamedUser.NamedUser attribute), 50
  - redirect (github.Download.Download attribute), 48
  - ref (github.GitRef.GitRef attribute), 92
  - ref (github.PullRequestPart.PullRequestPart attribute), 54
  - remaining (github.Rate.Rate attribute), 78
  - remove\_from\_assignees() (github.Issue.Issue method), 37
  - remove\_from\_collaborators() (github.Repository.Repository method), 77
  - remove\_from\_emails() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - remove\_from\_following() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - remove\_from\_labels() (github.Issue.Issue method), 37
  - remove\_from\_members() (github.Organization.Organization method), 60
  - remove\_from\_members() (github.Team.Team method), 39
  - remove\_from\_public\_members() (github.Organization.Organization method), 60
  - remove\_from\_repos() (github.Team.Team method), 39
  - remove\_from\_starred() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - remove\_from\_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 30
  - remove\_from\_watched() (github.AuthenticatedUser.AuthenticatedUser method), 31
  - render\_markdown() (github.MainClass.Github method), 6
  - repo (github.Event.Event attribute), 53
  - repo (github.PullRequestPart.PullRequestPart attribute), 54
  - repos\_count (github.Team.Team attribute), 38
  - repos\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - repos\_url (github.NamedUser.NamedUser attribute), 50
  - repos\_url (github.Organization.Organization attribute), 56
  - repositories\_url (github.Team.Team attribute), 38
  - Repository (class in github.Repository), 61
  - repository (github.CommitCombinedStatus.CommitCombinedStatus attribute), 85
  - repository (github.ContentFile.ContentFile attribute), 33
  - repository (github.Invitation.Invitation attribute), 44
  - repository (github.Issue.Issue attribute), 35
  - repository (github.Notification.Notification attribute), 95
  - repository\_url (github.SourceImport.SourceImport attribute), 91
  - RepositoryKey (class in github.RepositoryKey), 93
  - reset (github.Rate.Rate attribute), 78
  - user\_starred() (github.Gist.Gist method), 82
  - review\_comment\_url (github.PullRequest.PullRequest attribute), 97
  - review\_comments (github.PullRequest.PullRequest attribute), 97
  - review\_comments\_url (github.PullRequest.PullRequest attribute), 97
- ## S
- s3\_url (github.Download.Download attribute), 48
  - schema (github.HookDescription.HookDescription attribute), 44
  - scopes (github.Authorization.Authorization attribute), 101
  - search\_code() (github.MainClass.Github method), 5
  - search\_commits() (github.MainClass.Github method), 6
  - search\_issues() (github.MainClass.Github method), 5
  - search\_repositories() (github.MainClass.Github method), 5
  - search\_users() (github.MainClass.Github method), 5
  - set\_labels() (github.Issue.Issue method), 37
  - set\_repo\_permission() (github.Team.Team method), 38
  - set\_starred() (github.Gist.Gist method), 82
  - sha (github.Commit.Commit attribute), 33
  - sha (github.CommitCombinedStatus.CommitCombinedStatus attribute), 85
  - sha (github.ContentFile.ContentFile attribute), 33
  - sha (github.File.File attribute), 87
  - sha (github.GitBlob.GitBlob attribute), 60
  - sha (github.GitCommit.GitCommit attribute), 88
  - sha (github.GitObject.GitObject attribute), 102
  - sha (github.GitTag.GitTag attribute), 45
  - sha (github.GitTree.GitTree attribute), 101
  - sha (github.GitTreeElement.GitTreeElement attribute), 48
  - sha (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 86
  - sha (github.PullRequestPart.PullRequestPart attribute), 55
  - signature (github.Download.Download attribute), 48
  - site\_admin (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - site\_admin (github.NamedUser.NamedUser attribute), 51
  - size (github.ContentFile.ContentFile attribute), 33
  - size (github.Download.Download attribute), 48
  - size (github.GistFile.GistFile attribute), 84
  - size (github.GitBlob.GitBlob attribute), 61
  - size (github.GitReleaseAsset.GitReleaseAsset attribute), 85
  - size (github.GitTreeElement.GitTreeElement attribute), 48
  - size (github.Repository.Repository attribute), 64
  - slug (github.Team.Team attribute), 38

- source (github.GitignoreTemplate.GitignoreTemplate attribute), 53
- source (github.Repository.Repository attribute), 64
- SourceImport (class in github.SourceImport), 91
- space (github.Plan.Plan attribute), 83
- ssh\_url (github.Repository.Repository attribute), 64
- Stargazer (class in github.Stargazer), 92
- stargazers\_count (github.Repository.Repository attribute), 65
- stargazers\_url (github.Repository.Repository attribute), 65
- starred\_at (github.Stargazer.Stargazer attribute), 92
- starred\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 24
- starred\_url (github.NamedUser.NamedUser attribute), 51
- state (github.CommitCombinedStatus.CommitCombinedStatus attribute), 85
- state (github.CommitStatus.CommitStatus attribute), 42
- state (github.GitReleaseAsset.GitReleaseAsset attribute), 85
- state (github.Issue.Issue attribute), 35
- state (github.Milestone.Milestone attribute), 46
- state (github.PullRequest.PullRequest attribute), 97
- state (github.PullRequestReview.PullRequestReview attribute), 93
- stats (github.Commit.Commit attribute), 33
- StatsCodeFrequency (class in github.StatsCodeFrequency), 83
- StatsCommitActivity (class in github.StatsCommitActivity), 102
- StatsContributor (class in github.StatsContributor), 78
- StatsContributor.Week (class in github.StatsContributor), 78
- StatsParticipation (class in github.StatsParticipation), 84
- StatsPunchCard (class in github.StatsPunchCard), 47
- Status (class in github.Status), 61
- status (github.Comparison.Comparison attribute), 89
- status (github.File.File attribute), 87
- status (github.GithubException.GithubException attribute), 20
- status (github.HookResponse.HookResponse attribute), 54
- status (github.SourceImport.SourceImport attribute), 91
- status (github.Status.Status attribute), 61
- status (github.StatusMessage.StatusMessage attribute), 89
- status\_text (github.SourceImport.SourceImport attribute), 91
- statuses (github.CommitCombinedStatus.CommitCombinedStatus attribute), 86
- statuses\_url (github.Repository.Repository attribute), 65
- StatusMessage (class in github.StatusMessage), 89
- subject (github.Notification.Notification attribute), 95
- submitted\_at (github.PullRequestReview.PullRequestReview attribute), 93
- subscribe\_to\_hub() (github.Repository.Repository method), 78
- subscribers\_count (github.Repository.Repository attribute), 65
- subscribers\_url (github.Repository.Repository attribute), 65
- subscription\_url (github.Notification.Notification attribute), 95
- subscription\_url (github.Repository.Repository attribute), 65
- subscriptions\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 24
- subscriptions\_url (github.NamedUser.NamedUser attribute), 51
- supported\_events (github.HookDescription.HookDescription attribute), 44
- svn\_url (github.Repository.Repository attribute), 65

## T

- Tag (class in github.Tag), 40
- tag (github.GitTag.GitTag attribute), 45
- tag\_name (github.GitRelease.GitRelease attribute), 94
- tagger (github.GitTag.GitTag attribute), 45
- tags\_url (github.Repository.Repository attribute), 65
- tarball\_url (github.GitRelease.GitRelease attribute), 95
- tarball\_url (github.Tag.Tag attribute), 40
- target\_url (github.CommitStatus.CommitStatus attribute), 42
- Team (class in github.Team), 37
- teams\_url (github.Repository.Repository attribute), 65
- test() (github.Hook.Hook method), 41
- test\_url (github.Hook.Hook attribute), 41
- title (github.GitRelease.GitRelease attribute), 94
- title (github.Issue.Issue attribute), 35
- title (github.Milestone.Milestone attribute), 46
- title (github.NotificationSubject.NotificationSubject attribute), 100
- title (github.PullRequest.PullRequest attribute), 97
- title (github.RepositoryKey.RepositoryKey attribute), 93
- title (github.UserKey.UserKey attribute), 43
- token (github.Authorization.Authorization attribute), 101
- token (github.InstallationAuthorization.InstallationAuthorization attribute), 79
- total (github.CommitStats.CommitStats attribute), 102
- total (github.StatsCommitActivity.StatsCommitActivity attribute), 102
- total (github.StatsContributor.StatsContributor attribute), 79
- total\_commits (github.Comparison.Comparison attribute), 89
- total\_count (github.CommitCombinedStatus.CommitCombinedStatus attribute), 85

- total\_private\_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 24
  - total\_private\_repos (github.NamedUser.NamedUser attribute), 51
  - total\_private\_repos (github.Organization.Organization attribute), 56
  - transformation\_exception (github.GithubException.BadAttributeException attribute), 21
  - tree (github.GitCommit.GitCommit attribute), 88
  - tree (github.GitTree.GitTree attribute), 101
  - trees\_url (github.Repository.Repository attribute), 65
  - TwoFactorException, 21
  - type (github.AuthenticatedUser.AuthenticatedUser attribute), 25
  - type (github.ContentFile.ContentFile attribute), 33
  - type (github.Event.Event attribute), 53
  - type (github.GistFile.GistFile attribute), 84
  - type (github.GitObject.GitObject attribute), 102
  - type (github.GitTreeElement.GitTreeElement attribute), 48
  - type (github.NamedUser.NamedUser attribute), 51
  - type (github.NotificationSubject.NotificationSubject attribute), 100
  - type (github.Organization.Organization attribute), 56
- ## U
- UnknownObjectException, 21
  - unread (github.Notification.Notification attribute), 95
  - unsubscribe\_from\_hub() (github.Repository.Repository method), 78
  - update\_asset() (github.GitReleaseAsset.GitReleaseAsset method), 85
  - update\_file() (github.Repository.Repository method), 70
  - updated\_at (github.AuthenticatedUser.AuthenticatedUser attribute), 25
  - updated\_at (github.Authorization.Authorization attribute), 101
  - updated\_at (github.CommitComment.CommitComment attribute), 82
  - updated\_at (github.CommitStatus.CommitStatus attribute), 42
  - updated\_at (github.Gist.Gist attribute), 81
  - updated\_at (github.GistComment.GistComment attribute), 79
  - updated\_at (github.GistHistoryState.GistHistoryState attribute), 88
  - updated\_at (github.GitReleaseAsset.GitReleaseAsset attribute), 85
  - updated\_at (github.Hook.Hook attribute), 41
  - updated\_at (github.Issue.Issue attribute), 35
  - updated\_at (github.IssueComment.IssueComment attribute), 90
  - updated\_at (github.Milestone.Milestone attribute), 46
  - updated\_at (github.NamedUser.NamedUser attribute), 51
  - updated\_at (github.Notification.Notification attribute), 95
  - url (github.AuthenticatedUser.AuthenticatedUser attribute), 25
  - url (github.Authorization.Authorization attribute), 101
  - url (github.AuthorizationApplication.AuthorizationApplication attribute), 40
  - url (github.Commit.Commit attribute), 33
  - url (github.CommitCombinedStatus.CommitCombinedStatus attribute), 85
  - url (github.CommitComment.CommitComment attribute), 82
  - url (github.CommitStatus.CommitStatus attribute), 42
  - url (github.Comparison.Comparison attribute), 89
  - url (github.ContentFile.ContentFile attribute), 33
  - url (github.Download.Download attribute), 48
  - url (github.Gist.Gist attribute), 81
  - url (github.GistComment.GistComment attribute), 79
  - url (github.GistHistoryState.GistHistoryState attribute), 88
  - url (github.GitBlob.GitBlob attribute), 61
  - url (github.GitCommit.GitCommit attribute), 88
  - url (github.GitObject.GitObject attribute), 102
  - url (github.GitRef.GitRef attribute), 92
  - url (github.GitRelease.GitRelease attribute), 94
  - url (github.GitReleaseAsset.GitReleaseAsset attribute), 84
  - url (github.GitTag.GitTag attribute), 45
  - url (github.GitTree.GitTree attribute), 101
  - url (github.GitTreeElement.GitTreeElement attribute), 48
  - url (github.Hook.Hook attribute), 41
  - url (github.Invitation.Invitation attribute), 44
  - url (github.Issue.Issue attribute), 35
  - url (github.IssueComment.IssueComment attribute), 90
  - url (github.IssueEvent.IssueEvent attribute), 43
  - url (github.Label.Label attribute), 44
  - url (github.Milestone.Milestone attribute), 46
  - url (github.NamedUser.NamedUser attribute), 51
  - url (github.Notification.Notification attribute), 95
  - url (github.NotificationSubject.NotificationSubject attribute), 100
  - url (github.Organization.Organization attribute), 57
  - url (github.PullRequest.PullRequest attribute), 97
  - url (github.PullRequestComment.PullRequestComment attribute), 32



url (github.PullRequestReview.PullRequestReview attribute), 93

url (github.Repository.Repository attribute), 65

url (github.RepositoryKey.RepositoryKey attribute), 94

url (github.SourceImport.SourceImport attribute), 91

url (github.Team.Team attribute), 38

url (github.UserKey.UserKey attribute), 43

use\_ifs (github.SourceImport.SourceImport attribute), 91

user (github.CommitComment.CommitComment attribute), 82

user (github.Gist.Gist attribute), 81

user (github.GistComment.GistComment attribute), 79

user (github.GistHistoryState.GistHistoryState attribute), 88

user (github.Issue.Issue attribute), 36

user (github.IssueComment.IssueComment attribute), 90

user (github.PullRequest.PullRequest attribute), 98

user (github.PullRequestComment.PullRequestComment attribute), 32

user (github.PullRequestPart.PullRequestPart attribute), 55

user (github.PullRequestReview.PullRequestReview attribute), 93

user (github.Reaction.Reaction attribute), 54

user (github.Stargazer.Stargazer attribute), 92

UserKey (class in github.UserKey), 43

## V

vcs (github.SourceImport.SourceImport attribute), 91

vcs\_url (github.SourceImport.SourceImport attribute), 91

verified (github.RepositoryKey.RepositoryKey attribute), 94

verified (github.UserKey.UserKey attribute), 43

version (github.GistHistoryState.GistHistoryState attribute), 88

## W

w (github.StatsContributor.StatsContributor.Week attribute), 78

watchers (github.Repository.Repository attribute), 65

watchers\_count (github.Repository.Repository attribute), 65

week (github.StatsCodeFrequency.StatsCodeFrequency attribute), 83

week (github.StatsCommitActivity.StatsCommitActivity attribute), 102

weeks (github.StatsContributor.StatsContributor attribute), 79

## Z

zipball\_url (github.GitRelease.GitRelease attribute), 95

zipball\_url (github.Tag.Tag attribute), 40