

---

# **PyGithub Documentation**

***Release 1.25.2***

**Vincent Jacques**

August 03, 2015



<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	(Very short) tutorial . . . . .	1
1.2	Download and install . . . . .	1
1.3	Licensing . . . . .	1
1.4	What next? . . . . .	1
1.5	Projects using PyGithub . . . . .	2
1.6	They talk about PyGithub . . . . .	2
<b>2</b>	<b>Reference</b>	<b>3</b>
2.1	Main class: Github . . . . .	3
2.2	APIs . . . . .	7
2.3	Utilities . . . . .	19
2.4	Github objects . . . . .	21
<b>3</b>	<b>Change log</b>	<b>89</b>
3.1	Stable versions . . . . .	89
3.2	Pre-release versions . . . . .	95
<b>4</b>	<b>Contributing</b>	<b>99</b>
4.1	Issues . . . . .	99
4.2	Pull requests . . . . .	99
	<b>Python Module Index</b>	<b>101</b>



---

## Introduction

---

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

g = Github("user", "password")
```

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 `easy_install PyGithub` or `pip install PyGithub` should be enough. You can also clone it on [Github](#).

### 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)

- [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>

## 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)

---

## Reference

---

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)
```

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\_resettme**

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*)

**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`

**legacy\_search\_repos** (*keyword, language=NotSet*)

**Calls** `GET /legacy/repos/search/:keyword`



**Parameters**

- **keyword** – string
- **language** – string

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

**legacy\_search\_users** (*keyword*)

Calls `GET /legacy/user/search/:keyword`

**Parameters** **keyword** – string

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

**legacy\_search\_user\_by\_email** (*email*)

Calls `GET /legacy/user/email/:email`

**Parameters** **email** – string

**Return type** `github.NamedUser.NamedUser`

**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`

**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**

- **class** – the class of the object to create
- **raw\_data** – dict
- **headers** – dict

**Return type** instance of class `class`

**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`

## 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()`
- /issues
  - GET: `github.AuthenticatedUser.AuthenticatedUser.get_issues()`
- /legacy/issues/search/:owner/:repository/:state/:keyword
  - GET: `github.Repository.Repository.legacy_search_issues()`
- /legacy/repos/search/:keyword
  - GET: `github.MainClass.Github.legacy_search_repos()`
- /legacy/user/email/:email
  - GET: `github.MainClass.Github.legacy_search_user_by_email()`
- /legacy/user/search/:keyword
  - GET: `github.MainClass.Github.legacy_search_users()`
- /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()`
- `/rate_limit`
  - GET: Not implemented, see *Github.rate\_limiting*
- `/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()`
- `/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/commits
  - GET: `github.Repository.Repository.get_commits()`
- /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()`
- /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/tests`
  - POST: `github.Hook.Hook.test()`
- `/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/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/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_comments()`
  - PATCH: `github.IssueComment.IssueComment.edit()`
  - DELETE: `github.IssueComment.IssueComment.delete()`
- /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()`
  - PATCH: `github.RepositoryKey.RepositoryKey.edit()`
  - 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/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/readme`
  - GET: `github.Repository.Repository.get_readme()`
- `/repos/:owner/:repo/stargazers`
  - GET: `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/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/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/repos
  - GET: `github.Team.Team.get_repos()`
- /teams/:id/repos/:org/:repo
  - PUT: `github.Team.Team.add_to_repos()`
- /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/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()`
- `/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()`
- `/user/watched`
  - GET: `github.AuthenticatedUser.AuthenticatedUser.get_watched()`
- `/user/watched/:owner/:repo`
  - GET: `github.AuthenticatedUser.AuthenticatedUser.has_in_watched()`
  - PUT: `github.AuthenticatedUser.AuthenticatedUser.add_to_watched()`
  - DELETE: `github.AuthenticatedUser.AuthenticatedUser.remove_from_watched()`

- /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)`

### Parameters

- **content** – string
- **new\_name** – string

`class github.InputGitAuthor.InputGitAuthor (name, email, date)`

### Parameters

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

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

### 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

### 2.4.1 IssueEvent

**class** `github.IssueEvent.IssueEvent`

This class represents IssueEvents as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.2 StatsContributor

**class** `github.StatsContributor.StatsContributor`

This class represents statistics of a contributor. 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

StatsContributor.author
    Type github.NamedUser.NamedUser

StatsContributor.total
    Type int

StatsContributor.weeks
    Type list of Week
```

### 2.4.3 PullRequestPart

```
class github.PullRequestPart.PullRequestPart
    This class represents PullRequestParts as returned for example by http://developer.github.com/v3/todo

    label
        Type string

    ref
        Type string

    repo
        Type github.Repository.Repository

    sha
        Type string

    user
        Type github.NamedUser.NamedUser
```

### 2.4.4 HookResponse

```
class github.HookResponse.HookResponse
    This class represents HookResponses as returned for example by http://developer.github.com/v3/todo

    code
        Type integer

    message
        Type string

    status
        Type string
```

## 2.4.5 CommitComment

**class** `github.CommitComment.CommitComment`

This class represents CommitComments as returned for example by <http://developer.github.com/v3/todo>

**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

## 2.4.6 CommitStatus

**class** `github.CommitStatus.CommitStatus`

This class represents CommitStatus as returned for example by <http://developer.github.com/v3/todo>

**created\_at**

**Type** `datetime.datetime`

**creator**

**Type** `github.NamedUser.NamedUser`

**description**

**Type** `string`

**id**

**Type** `integer`

**state**

**Type** `string`

**target\_url**

**Type** `string`

**updated\_at**

**Type** `datetime.datetime`

**url**

**Type** `string`

## 2.4.7 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\_repo** (*name*, *description=NotSet*, *homepage=NotSet*, *private=NotSet*, *has\_issues=NotSet*, *has\_wiki=NotSet*, *has\_downloads=NotSet*, *team\_id=NotSet*, *auto\_init=NotSet*, *gitignore\_template=NotSet*)

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

**Parameters**

- **name** – string
- **description** – string
- **homepage** – string
- **private** – bool
- **has\_issues** – bool
- **has\_wiki** – bool
- **has\_downloads** – bool
- **team\_id** – *github.Team.Team*
- **auto\_init** – bool
- **gitignore\_template** – string

**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*

**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*

**get\_events** ()

**Calls** *GET /orgs/:org/events*

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

**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** ()

**Calls** *GET /orgs/:org/members*

**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

**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

## 2.4.8 ContentFile

**class** `github.ContentFile.ContentFile`

This class represents ContentFiles as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.9 GistComment

**class** *github.GistComment.GistComment*

This class represents GistComments as returned for example by <http://developer.github.com/v3/todo>

**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`

## 2.4.10 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`

**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*

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

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

**Parameters**

- **title** – string
- **body** – string
- **state** – 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** ()

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

**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`

**is\_merged** ()

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

**Return type** `bool`

**merge** (*commit\_message=NotSet*)

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

**Parameters** `commit_message` – string

**Return type** `github.PullRequestMergeStatus.PullRequestMergeStatus`

### 2.4.11 Plan

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

This class represents Plans as returned for example by <http://developer.github.com/v3/todo>

**collaborators**

**Type** integer

**name**

**Type** string

**private\_repos**

**Type** integer

**space**

**Type** integer

### 2.4.12 NotificationSubject

**class** `github.NotificationSubject.NotificationSubject`

This class represents Subjects of Notifications as returned for example by <http://developer.github.com/v3/activity/notifications/#list-your-notifications>

**title**

**Type** string

**url**

**Type** string

**latest\_comment\_url**

**Type** string

**type**

**Type** string

### 2.4.13 GitAuthor

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

This class represents GitAuthors as returned for example by <http://developer.github.com/v3/todo>

**date**

**Type** `datetime.datetime`

**email**

**Type** string

**name**

**Type** string

### 2.4.14 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`

### 2.4.15 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*)

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

Parameters

- **state** – string
- **target\_url** – string
- **description** – string

Return type *github.CommitStatus.CommitStatus*

**get\_comments** ()

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

Return type *github.PaginatedList.PaginatedList*  
*github.CommitComment.CommitComment*

of

**get\_statuses** ()

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

Return type *github.PaginatedList.PaginatedList*  
*github.CommitStatus.CommitStatus*

of

## 2.4.16 GitBlob

class *github.GitBlob.GitBlob*

This class represents GitBlobs as returned for example by <http://developer.github.com/v3/todo>

**content**

Type string

**encoding**

Type string



**sha**  
Type string

**size**  
Type integer

**url**  
Type string

### 2.4.17 File

**class** `github.File.File`

This class represents Files as returned for example by <http://developer.github.com/v3/todo>

**additions**  
Type integer

**blob\_url**  
Type string

**changes**  
Type integer

**contents\_url**  
Type string

**deletions**  
Type integer

**filename**  
Type string

**patch**  
Type string

**raw\_url**  
Type string

**sha**  
Type string

**status**  
Type string

### 2.4.18 IssueComment

**class** `github.IssueComment.IssueComment`

This class represents IssueComments as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.19 Repository

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

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

**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

**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*)

**Calls** [PUT /repos/:owner/:repo/collaborators/:user](#)

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

**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*)

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

Parameters

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

Return type *github.Issue.Issue*

**create\_key** (*title*, *key*)

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

Parameters

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

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** – date

Return type *github.Milestone.Milestone*

**create\_pull** (*\*args*, *\*\*kws*)

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



**Parameters**

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

**Return type** *github.PullRequest.PullRequest*

**delete()**

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

**Return type** None

**edit** (*name*, *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*  
*github.NamedUser.NamedUser*

of

**get\_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`

**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*)

**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*

**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*)**  
Calls `GET /repos/:owner/:repo/pulls`  
Parameters **state** – 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\_stargazers** ()

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

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

**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\_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`

**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

## 2.4.20 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`

## 2.4.21 GitRef

**class** `github.GitRef.GitRef`

This class represents GitRefs as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.22 AuthorizationApplication

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

This class represents AuthorizationApplications as returned for example by <http://developer.github.com/v3/todo>

**name**

**Type** string

**url**

**Type** string

### 2.4.23 CommitStats

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

This class represents CommitStatss as returned for example by <http://developer.github.com/v3/todo>

**additions**

**Type** integer

**deletions**

**Type** integer

**total**

**Type** integer

### 2.4.24 GitTree

**class** github.GitTree.**GitTree**

This class represents GitTrees as returned for example by <http://developer.github.com/v3/todo>

**sha**

**Type** string

**tree**  
 Type list of *github.GitTreeElement.GitTreeElement*

**url**  
 Type string

### 2.4.25 GitCommit

**class** `github.GitCommit.GitCommit`

This class represents GitCommits as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.26 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

### 2.4.27 Hook

**class** *github.Hook.Hook*

This class represents Hooks as returned for example by <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
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

```

### 2.4.28 IssuePullRequest

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

This class represents IssuePullRequests as returned for example by <http://developer.github.com/v3/todo>

```
    diff_url
```

**Type** string  
**html\_url**  
**Type** string  
**patch\_url**  
**Type** string

### 2.4.29 HookDescription

**class** `github.HookDescription.HookDescription`

This class represents HookDescriptions as returned for example by <http://developer.github.com/v3/todo>

**events**  
**Type** list of string  
**name**  
**Type** string  
**schema**  
**Type** list of list of string  
**supported\_events**  
**Type** list of string

### 2.4.30 RateLimit

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

This class represents rate limits as defined in [http://developer.github.com/v3/rate\\_limit](http://developer.github.com/v3/rate_limit)

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

### 2.4.31 GitTreeElement

**class** `github.GitTreeElement.GitTreeElement`

This class represents GitTreeElements as returned for example by <http://developer.github.com/v3/todo>

**mode**  
**Type** string  
**path**  
**Type** string  
**sha**  
**Type** string  
**size**  
**Type** integer  
**type**

**Type** string

**url**

**Type** string

### 2.4.32 RepositoryKey

**class** github.RepositoryKey.**RepositoryKey**

This class represents RepositoryKeys.  
<http://developer.github.com/v3/repos/keys/>

The reference can be found here

**id**

**Type** integer

**key**

**Type** string

**title**

**Type** string

**url**

**Type** string

**verified**

**Type** bool

**delete()**

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

**Return type** None

**edit** (*title=NotSet, key=NotSet*)

**Calls** `PATCH /repos/:owner/:repo/keys/:id`

**Parameters**

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

**Return type** None

### 2.4.33 AuthenticatedUser

**class** github.AuthenticatedUser.**AuthenticatedUser**

This class represents AuthenticatedUsers as returned for example by <http://developer.github.com/v3/todo>

**avatar\_url**

**Type** string

**bio**

**Type** string

**blog**

**Type** string

**collaborators**  
    Type integer

**company**  
    Type string

**created\_at**  
    Type datetime.datetime

**disk\_usage**  
    Type integer

**email**  
    Type string

**events\_url**  
    Type string

**followers**  
    Type integer

**followers\_url**  
    Type string

**following**  
    Type integer

**following\_url**  
    Type string

**gists\_url**  
    Type string

**gravatar\_id**  
    Type string

**hireable**  
    Type bool

**html\_url**  
    Type string

**id**  
    Type integer

**location**  
    Type string

**login**  
    Type string

**name**  
    Type string



**organizations\_url**  
Type string

**owned\_private\_repos**  
Type integer

**plan**  
Type *github.Plan.Plan*

**private\_gists**  
Type integer

**public\_gists**  
Type integer

**public\_repos**  
Type integer

**received\_events\_url**  
Type string

**repos\_url**  
Type string

**site\_admin**  
Type bool

**starred\_url**  
Type string

**subscriptions\_url**  
Type string

**total\_private\_repos**  
Type integer

**type**  
Type string

**updated\_at**  
Type datetime.datetime

**url**  
Type string

**add\_to\_emails** (\*emails)  
Calls [POST /user/emails](#)  
Parameters **email** – string  
Return type None

**add\_to\_following** (following)  
Calls [PUT /user/following/:user](#)

Parameters following – *github.NamedUser.NamedUser*

Return type None

**add\_to\_starred** (*starred*)

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

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

Return type None

**add\_to\_subscriptions** (*subscription*)

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

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

Return type None

**add\_to\_watched** (*watched*)

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

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*, *auto\_init=NotSet*, *gitignore\_template=NotSet*)

**Calls** `POST /user/repos`

**Parameters**

- **name** – string
- **description** – string
- **homepage** – string
- **private** – bool
- **has\_issues** – bool
- **has\_wiki** – bool
- **has\_downloads** – bool
- **auto\_init** – bool
- **gitignore\_template** – string

**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/watched`

**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 /user/watched/:owner/:repo`

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 /user/watched/:owner/:repo`

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

Return type `None`

## 2.4.34 StatsParticipation

**class** `github.StatsParticipation.StatsParticipation`

This class represents statistics of participation. 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`

### 2.4.35 GistHistoryState

**class** `github.GistHistoryState.GistHistoryState`

This class represents GistHistoryStates as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.36 GitObject

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

This class represents GitObjects as returned for example by <http://developer.github.com/v3/todo>

**sha**  
**Type** string  
**type**  
**Type** string  
**url**  
**Type** string

### 2.4.37 Download

**class** `github.Download.Download`

This class represents Downloads as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.38 NamedUser

**class** `github.NamedUser`.**NamedUser**

This class represents NamedUsers as returned for example by <http://developer.github.com/v3/todo>

**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

**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

**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)**

**Calls** GET /users/:user/repos

**Parameters** **type** – 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`

## 2.4.39 Permissions

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

This class represents Permissions as returned for example by <http://developer.github.com/v3/todo>

**admin**  
Type `bool`

**pull**  
Type `bool`

**push**  
Type `bool`

## 2.4.40 Issue

**class** `github.Issue.Issue`

This class represents Issues as returned for example by <http://developer.github.com/v3/todo>

**assignee**  
Type `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\_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*)

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

Parameters

- **title** – string
- **body** – string
- **assignee** – string or `github.NamedUser.NamedUser` or None
- **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** ()

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

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\_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

### 2.4.41 StatusMessage

**class** `github.StatusMessage.StatusMessage`

This class represents status messages as defined in <https://status.github.com/api>

**body**

**Type** string

**status**

**Type** string

**created\_on**

**Type** `datetime.datetime`

### 2.4.42 Authorization

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

This class represents Authorizations as returned for example by <http://developer.github.com/v3/todo>

**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

### 2.4.43 Branch

**class** `github.Branch.Branch`

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

**commit**

**Type** `github.Commit.Commit`

**name**

**Type** string

### 2.4.44 Rate

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

This class represents rate limits as defined in [http://developer.github.com/v3/rate\\_limit](http://developer.github.com/v3/rate_limit)

**limit**

**Type** integer

**remaining**

**Type** integer

**reset**

**Type** `datetime.datetime`

### 2.4.45 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

### 2.4.46 StatsCommitActivity

**class** `github.StatsCommitActivity.StatsCommitActivity`

This class represents statistics of commit activity. 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

### 2.4.47 GitTag

**class** `github.GitTag.GitTag`

This class represents GitTags as returned for example by <http://developer.github.com/v3/todo>

**message**

**Type** `string`

**object**

**Type** `github.GitObject.GitObject`

**sha**

**Type** `string`

**tag**

**Type** `string`

**tagger**

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

**url**

**Type** `string`

### 2.4.48 Comparison

**class** `github.Comparison.Comparison`

This class represents Comparisons as returned for example by <http://developer.github.com/v3/todo>

**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

```

#### 2.4.49 GistFile

**class** `github.GistFile.GistFile`

This class represents GistFiles as returned for example by <http://developer.github.com/v3/todo>

```

    content
        Type string
    filename
        Type string
    language
        Type string
    raw_url
        Type string
    size
        Type integer
    type
        Type string

```

#### 2.4.50 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
```

### 2.4.51 Status

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

This class represents status as defined in <https://status.github.com/api>

**status**

**Type** string

**last\_updated**

**Type** datetime.datetime

### 2.4.52 Gist

**class** `github.Gist.Gist`

This class represents Gists as returned for example by <http://developer.github.com/v3/todo>

**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`

### 2.4.53 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`

### 2.4.54 StatsCodeFrequency

**class** `github.StatsCodeFrequency.StatsCodeFrequency`

This class represents statistics of code frequency. 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`

### 2.4.55 PullRequestMergeStatus

**class** `github.PullRequestMergeStatus.PullRequestMergeStatus`

This class represents PullRequestMergeStatus. 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`

### 2.4.56 StatsPunchCard

**class** `github.StatsPunchCard.StatsPunchCard`

This class represents the punch card. 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

### 2.4.57 Tag

**class** `github.Tag.Tag`

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

**commit**

**Type** `github.Commit.Commit`

**name**

**Type** `string`

**tarball\_url**

**Type** `string`

**zipball\_url**

Type string

## 2.4.58 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

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

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

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

Return type None

**add\_to\_repos** (*repo*)

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

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

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** ()

Calls `GET /teams/:id/members`

**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

## 2.4.59 GitignoreTemplate

**class** `github.GitignoreTemplate.GitignoreTemplate`

This class represents GitignoreTemplates as returned for example by <http://developer.github.com/v3/todo>

**source**

**Type** string

**name**

**Type** string

---

**Change log**

---

## **3.1 Stable versions**

### **3.1.1 Version 1.25.2 (October 7th, 2014)**

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

### **3.1.2 Version 1.25.1 (September 28th, 2014)**

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

### **3.1.3 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

### **3.1.4 Version 1.24.1 (March 16th, 2014)**

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

### **3.1.5 Version 1.24.0 (March 2nd, 2014)**

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

### **3.1.6 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](#)

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

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

### 3.1.8 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

### 3.1.9 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

### 3.1.10 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)

### 3.1.11 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_resettime`. 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

### 3.1.12 Version 1.17.0 (Jully 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](#)

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

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

### 3.1.14 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

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

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

### 3.1.16 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.

### 3.1.17 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

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

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

### 3.1.19 Version 1.13.0 (March 22nd, 2013)

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

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

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

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

- Nothing, but packaging/upload of 1.12.0 failed

### 3.1.22 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

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

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

### 3.1.24 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



### 3.1.25 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()`)

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

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

### 3.1.27 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

### 3.1.28 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](#)

### 3.1.29 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](#)

### 3.1.30 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)

### 3.1.31 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

### 3.1.32 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

### 3.1.33 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

### 3.1.34 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

### 3.1.35 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

### 3.1.36 Version 1.1 (June 20th, 2012)

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

### 3.1.37 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



---

## Contributing

---

### 4.1 Issues

A good issue includes a [short, self contained, correct example](#) of the problem, something like:

```
assert github.Github().get_user("jacquev6").name == "Vincent Jacques"
```

It is even better if you provide the debug logs associated with your issue. Enable them with `github.enable_console_debug_logging()` and copy them in the body of the issue. Warning, you may want to remove some private information (authentication information is removed, but there may be private stuff in the messages)

If for any reason you are not able to do that, open your issue anyway and we will see what is needed to solve your problem.

### 4.2 Pull requests

Please do your pull requests on the `develop` branch.

#### 4.2.1 Automated tests

You can run the tests through `python -m github.tests`. Run a specific test with `python -m github.tests TestCase` or `python -m github.tests TestCase.testMethod`.

If you add a new test, for example `Issue139.testCompletion`, you must add an `import` in `github/tests/AllTests.py`. Then, you have to run `python -m github.tests Issue139.testCompletion --record` to create the `github/tests/ReplayData/*.txt` files needed for you new test. Check them and commit them as well. You will need a `GithubCredentials.py` file at the root of the project with the following contents:

```
login = "my_login"
password = "my_password"
oauth_token = "my_token" # Can be left empty if not used
```

#### 4.2.2 Coding conventions

PyGithub follows [pep8 Style Guide for Python Code](#) except for line length. Please check your code with [pep8 Python style guide checker](#), by running `pep8 --ignore=E501 github`.





## g

`github`, [3](#)

`github.GithubException`, [19](#)



## A

- a (github.StatsContributor.StatsContributor.Week attribute), 22
- accesskeyid (github.Download.Download attribute), 69
- acl (github.Download.Download attribute), 69
- active (github.Hook.Hook attribute), 56
- actor (github.Event.Event attribute), 53
- actor (github.IssueEvent.IssueEvent attribute), 21
- actual\_value (github.GithubException.BadAttributeException attribute), 19
- add\_to\_collaborators() (github.Repository.Repository method), 42
- add\_to\_emails() (github.AuthenticatedUser.AuthenticatedUser method), 61
- add\_to\_following() (github.AuthenticatedUser.AuthenticatedUser method), 61
- add\_to\_labels() (github.Issue.Issue method), 75
- add\_to\_members() (github.Team.Team method), 87
- add\_to\_public\_members() (github.Organization.Organization method), 26
- add\_to\_repos() (github.Team.Team method), 87
- add\_to\_starred() (github.AuthenticatedUser.AuthenticatedUser method), 62
- add\_to\_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 62
- add\_to\_watched() (github.AuthenticatedUser.AuthenticatedUser method), 62
- additions (github.CommitStats.CommitStats attribute), 54
- additions (github.File.File attribute), 37
- additions (github.PullRequest.PullRequest attribute), 30
- additions (github.StatsCodeFrequency.StatsCodeFrequency attribute), 86
- admin (github.Permissions.Permissions attribute), 74
- ahead\_by (github.Comparison.Comparison attribute), 80
- all (github.StatsParticipation.StatsParticipation attribute), 67
- app (github.Authorization.Authorization attribute), 77
- archive\_url (github.Repository.Repository attribute), 38
- assignee (github.Issue.Issue attribute), 74
- assignee (github.PullRequest.PullRequest attribute), 30
- assignees\_url (github.Repository.Repository attribute), 38
- AuthenticatedUser (class in github.AuthenticatedUser), 59
- author (github.Commit.Commit attribute), 35
- author (github.GitCommit.GitCommit attribute), 55
- author (github.StatsContributor.StatsContributor attribute), 22
- Authorization (class in github.Authorization), 77
- AuthorizationApplication (class in github.AuthorizationApplication), 54
- avatar\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 59
- avatar\_url (github.NamedUser.NamedUser attribute), 70
- avatar\_url (github.Organization.Organization attribute), 24

## B

- BadAttributeException, 19
- BadCredentialsException, 19
- BadUserAgentException, 19
- base (github.PullRequest.PullRequest attribute), 30
- base\_commit (github.Comparison.Comparison attribute), 80
- behind\_by (github.Comparison.Comparison attribute), 80
- billing\_email (github.Organization.Organization attribute), 24
- bio (github.AuthenticatedUser.AuthenticatedUser attribute), 59
- bio (github.NamedUser.NamedUser attribute), 70
- blob\_url (github.File.File attribute), 37
- blobs\_url (github.Repository.Repository attribute), 38
- blog (github.AuthenticatedUser.AuthenticatedUser attribute), 59
- blog (github.NamedUser.NamedUser attribute), 71
- blog (github.Organization.Organization attribute), 24
- body (github.CommitComment.CommitComment attribute), 23
- body (github.GistComment.GistComment attribute), 29
- body (github.Issue.Issue attribute), 74
- body (github.IssueComment.IssueComment attribute), 37
- body (github.PullRequest.PullRequest attribute), 30

- body (github.PullRequestComment.PullRequestComment attribute), 55
  - body (github.StatusMessage.StatusMessage attribute), 77
  - Branch (class in github.Branch), 78
  - branches\_url (github.Repository.Repository attribute), 38
  - bucket (github.Download.Download attribute), 69
- ## C
- c (github.StatsContributor.StatsContributor.Week attribute), 22
  - change\_status (github.GistHistoryState.GistHistoryState attribute), 68
  - changed\_files (github.PullRequest.PullRequest attribute), 30
  - changes (github.File.File attribute), 37
  - clone\_url (github.Repository.Repository attribute), 38
  - closed\_at (github.Issue.Issue attribute), 74
  - closed\_at (github.PullRequest.PullRequest attribute), 30
  - closed\_by (github.Issue.Issue attribute), 74
  - closed\_issues (github.Milestone.Milestone attribute), 81
  - code (github.HookResponse.HookResponse attribute), 22
  - collaborators (github.AuthenticatedUser.AuthenticatedUser attribute), 60
  - collaborators (github.NamedUser.NamedUser attribute), 71
  - collaborators (github.Organization.Organization attribute), 24
  - collaborators (github.Plan.Plan attribute), 34
  - collaborators\_url (github.Repository.Repository attribute), 38
  - color (github.Label.Label attribute), 35
  - comments (github.Gist.Gist attribute), 83
  - comments (github.GistHistoryState.GistHistoryState attribute), 68
  - comments (github.Issue.Issue attribute), 75
  - comments (github.PullRequest.PullRequest attribute), 30
  - comments\_url (github.Commit.Commit attribute), 35
  - comments\_url (github.Gist.Gist attribute), 83
  - comments\_url (github.GistHistoryState.GistHistoryState attribute), 68
  - comments\_url (github.Issue.Issue attribute), 75
  - comments\_url (github.PullRequest.PullRequest attribute), 30
  - comments\_url (github.Repository.Repository attribute), 39
  - Commit (class in github.Commit), 35
  - commit (github.Branch.Branch attribute), 78
  - commit (github.Commit.Commit attribute), 35
  - commit (github.Tag.Tag attribute), 86
  - commit\_id (github.CommitComment.CommitComment attribute), 23
  - commit\_id (github.IssueEvent.IssueEvent attribute), 21
  - commit\_id (github.PullRequestComment.PullRequestComment attribute), 55
  - CommitComment (class in github.CommitComment), 23
  - commits (github.Comparison.Comparison attribute), 80
  - commits (github.PullRequest.PullRequest attribute), 30
  - commits\_url (github.Gist.Gist attribute), 83
  - commits\_url (github.GistHistoryState.GistHistoryState attribute), 68
  - commits\_url (github.PullRequest.PullRequest attribute), 30
  - commits\_url (github.Repository.Repository attribute), 39
  - CommitStats (class in github.CommitStats), 54
  - CommitStatus (class in github.CommitStatus), 24
  - committed\_at (github.GistHistoryState.GistHistoryState attribute), 68
  - committer (github.Commit.Commit attribute), 35
  - committer (github.GitCommit.GitCommit attribute), 55
  - company (github.AuthenticatedUser.AuthenticatedUser attribute), 60
  - company (github.NamedUser.NamedUser attribute), 71
  - company (github.Organization.Organization attribute), 24
  - compare() (github.Repository.Repository method), 42
  - compare\_url (github.Repository.Repository attribute), 39
  - Comparison (class in github.Comparison), 80
  - config (github.Hook.Hook attribute), 56
  - content (github.ContentFile.ContentFile attribute), 28
  - content (github.GistFile.GistFile attribute), 81
  - content (github.GitBlob.GitBlob attribute), 36
  - content\_type (github.Download.Download attribute), 69
  - ContentFile (class in github.ContentFile), 28
  - contents\_url (github.File.File attribute), 37
  - contents\_url (github.Repository.Repository attribute), 39
  - contributions (github.NamedUser.NamedUser attribute), 71
  - contributors\_url (github.Repository.Repository attribute), 39
  - create\_authorization() (github.AuthenticatedUser.AuthenticatedUser method), 62
  - create\_comment() (github.Commit.Commit method), 36
  - create\_comment() (github.Gist.Gist method), 84
  - create\_comment() (github.Issue.Issue method), 76
  - create\_comment() (github.PullRequest.PullRequest method), 32
  - create\_fork() (github.AuthenticatedUser.AuthenticatedUser method), 62
  - create\_fork() (github.Gist.Gist method), 84
  - create\_fork() (github.Organization.Organization method), 26
  - create\_from\_raw\_data() (github.MainClass.Github method), 6
  - create\_gist() (github.AuthenticatedUser.AuthenticatedUser method), 62
  - create\_git\_blob() (github.Repository.Repository method), 42
  - create\_git\_commit() (github.Repository.Repository method), 43

- `create_git_ref()` (`github.Repository.Repository` method), 43
  - `create_git_tag()` (`github.Repository.Repository` method), 43
  - `create_git_tree()` (`github.Repository.Repository` method), 43
  - `create_hook()` (`github.Repository.Repository` method), 43
  - `create_issue()` (`github.Repository.Repository` method), 44
  - `create_issue_comment()` (`github.PullRequest.PullRequest` method), 32
  - `create_key()` (`github.AuthenticatedUser.AuthenticatedUser` method), 62
  - `create_key()` (`github.Repository.Repository` method), 44
  - `create_label()` (`github.Repository.Repository` method), 44
  - `create_milestone()` (`github.Repository.Repository` method), 44
  - `create_pull()` (`github.Repository.Repository` method), 44
  - `create_repo()` (`github.AuthenticatedUser.AuthenticatedUser` method), 63
  - `create_repo()` (`github.Organization.Organization` method), 26
  - `create_review_comment()` (`github.PullRequest.PullRequest` method), 32
  - `create_status()` (`github.Commit.Commit` method), 36
  - `create_team()` (`github.Organization.Organization` method), 26
  - `created_at` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60
  - `created_at` (`github.Authorization.Authorization` attribute), 77
  - `created_at` (`github.CommitComment.CommitComment` attribute), 23
  - `created_at` (`github.CommitStatus.CommitStatus` attribute), 24
  - `created_at` (`github.Download.Download` attribute), 69
  - `created_at` (`github.Event.Event` attribute), 53
  - `created_at` (`github.Gist.Gist` attribute), 83
  - `created_at` (`github.GistComment.GistComment` attribute), 29
  - `created_at` (`github.GistHistoryState.GistHistoryState` attribute), 68
  - `created_at` (`github.Hook.Hook` attribute), 56
  - `created_at` (`github.Issue.Issue` attribute), 75
  - `created_at` (`github.IssueComment.IssueComment` attribute), 37
  - `created_at` (`github.IssueEvent.IssueEvent` attribute), 21
  - `created_at` (`github.Milestone.Milestone` attribute), 81
  - `created_at` (`github.NamedUser.NamedUser` attribute), 71
  - `created_at` (`github.Organization.Organization` attribute), 24
  - `created_at` (`github.PullRequest.PullRequest` attribute), 30
  - `created_at` (`github.PullRequestComment.PullRequestComment` attribute), 55
  - `created_at` (`github.Repository.Repository` attribute), 39
  - `created_on` (`github.StatusMessage.StatusMessage` attribute), 77
  - `creator` (`github.CommitStatus.CommitStatus` attribute), 24
  - `creator` (`github.Milestone.Milestone` attribute), 81
- ## D
- `d` (`github.StatsContributor.StatsContributor.Week` attribute), 22
  - `data` (`github.GithubException.GithubException` attribute), 19
  - `date` (`github.GitAuthor.GitAuthor` attribute), 34
  - `days` (`github.StatsCommitActivity.StatsCommitActivity` attribute), 79
  - `default_branch` (`github.Repository.Repository` attribute), 39
  - `delete()` (`github.Authorization.Authorization` method), 78
  - `delete()` (`github.CommitComment.CommitComment` method), 23
  - `delete()` (`github.Download.Download` method), 70
  - `delete()` (`github.Gist.Gist` method), 84
  - `delete()` (`github.GistComment.GistComment` method), 29
  - `delete()` (`github.GitRef.GitRef` method), 54
  - `delete()` (`github.Hook.Hook` method), 57
  - `delete()` (`github.IssueComment.IssueComment` method), 38
  - `delete()` (`github.Label.Label` method), 35
  - `delete()` (`github.Milestone.Milestone` method), 82
  - `delete()` (`github.PullRequestComment.PullRequestComment` method), 56
  - `delete()` (`github.Repository.Repository` method), 45
  - `delete()` (`github.RepositoryKey.RepositoryKey` method), 59
  - `delete()` (`github.Team.Team` method), 87
  - `delete()` (`github.UserKey.UserKey` method), 79
  - `delete_labels()` (`github.Issue.Issue` method), 76
  - `deletions` (`github.CommitStats.CommitStats` attribute), 54
  - `deletions` (`github.File.File` attribute), 37
  - `deletions` (`github.PullRequest.PullRequest` attribute), 30
  - `deletions` (`github.StatsCodeFrequency.StatsCodeFrequency` attribute), 86
  - `description` (`github.CommitStatus.CommitStatus` attribute), 24
  - `description` (`github.Download.Download` attribute), 69
  - `description` (`github.Gist.Gist` attribute), 83
  - `description` (`github.GistHistoryState.GistHistoryState` attribute), 68
  - `description` (`github.Milestone.Milestone` attribute), 81
  - `description` (`github.Repository.Repository` attribute), 39
  - `diff_hunk` (`github.PullRequestComment.PullRequestComment` attribute), 55
  - `diff_url` (`github.Comparison.Comparison` attribute), 80

`diff_url` (`github.IssuePullRequest.IssuePullRequest` attribute), 57  
`diff_url` (`github.PullRequest.PullRequest` attribute), 30  
`disk_usage` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`disk_usage` (`github.NamedUser.NamedUser` attribute), 71  
`disk_usage` (`github.Organization.Organization` attribute), 24  
`Download` (class in `github.Download`), 69  
`download_count` (`github.Download.Download` attribute), 69  
`downloads_url` (`github.Repository.Repository` attribute), 39  
`due_on` (`github.Milestone.Milestone` attribute), 82  
`dump()` (`github.MainClass.Github` method), 7

## E

`edit()` (`github.AuthenticatedUser.AuthenticatedUser` method), 63  
`edit()` (`github.Authorization.Authorization` method), 78  
`edit()` (`github.CommitComment.CommitComment` method), 23  
`edit()` (`github.Gist.Gist` method), 84  
`edit()` (`github.GistComment.GistComment` method), 30  
`edit()` (`github.GitRef.GitRef` method), 54  
`edit()` (`github.Hook.Hook` method), 57  
`edit()` (`github.Issue.Issue` method), 76  
`edit()` (`github.IssueComment.IssueComment` method), 38  
`edit()` (`github.Label.Label` method), 35  
`edit()` (`github.Milestone.Milestone` method), 82  
`edit()` (`github.Organization.Organization` method), 27  
`edit()` (`github.PullRequest.PullRequest` method), 32  
`edit()` (`github.PullRequestComment.PullRequestComment` method), 56  
`edit()` (`github.Repository.Repository` method), 45  
`edit()` (`github.RepositoryKey.RepositoryKey` method), 59  
`edit()` (`github.Team.Team` method), 87  
`edit()` (`github.UserKey.UserKey` method), 79  
`email` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`email` (`github.GitAuthor.GitAuthor` attribute), 34  
`email` (`github.NamedUser.NamedUser` attribute), 71  
`email` (`github.Organization.Organization` attribute), 25  
`enable_console_debug_logging()` (in module `github`), 19  
`encoding` (`github.ContentFile.ContentFile` attribute), 29  
`encoding` (`github.GitBlob.GitBlob` attribute), 36  
`etag` (`github.GithubObject.GithubObject` attribute), 21  
`Event` (class in `github.Event`), 53  
`event` (`github.IssueEvent.IssueEvent` attribute), 21  
`events` (`github.Hook.Hook` attribute), 57  
`events` (`github.HookDescription.HookDescription` attribute), 58  
`events_url` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60

`events_url` (`github.Issue.Issue` attribute), 75  
`events_url` (`github.NamedUser.NamedUser` attribute), 71  
`events_url` (`github.Organization.Organization` attribute), 25  
`events_url` (`github.Repository.Repository` attribute), 39  
`expected_type` (`github.GithubException.BadAttributeException` attribute), 19  
`expirationdate` (`github.Download.Download` attribute), 70

## F

`File` (class in `github.File`), 37  
`filename` (`github.File.File` attribute), 37  
`filename` (`github.GistFile.GistFile` attribute), 81  
`files` (`github.Commit.Commit` attribute), 35  
`files` (`github.Comparison.Comparison` attribute), 80  
`files` (`github.Gist.Gist` attribute), 83  
`files` (`github.GistHistoryState.GistHistoryState` attribute), 68  
`FIX_REPO_GET_GIT_REF` (`github.MainClass.Github` attribute), 3  
`followers` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`followers` (`github.NamedUser.NamedUser` attribute), 71  
`followers` (`github.Organization.Organization` attribute), 25  
`followers_url` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`followers_url` (`github.NamedUser.NamedUser` attribute), 71  
`following` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`following` (`github.NamedUser.NamedUser` attribute), 71  
`following` (`github.Organization.Organization` attribute), 25  
`following_url` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`following_url` (`github.NamedUser.NamedUser` attribute), 71  
`fork` (`github.Repository.Repository` attribute), 39  
`fork_of` (`github.Gist.Gist` attribute), 83  
`forks` (`github.Gist.Gist` attribute), 83  
`forks` (`github.GistHistoryState.GistHistoryState` attribute), 68  
`forks` (`github.Repository.Repository` attribute), 39  
`forks_count` (`github.Repository.Repository` attribute), 39  
`forks_url` (`github.Gist.Gist` attribute), 83  
`forks_url` (`github.GistHistoryState.GistHistoryState` attribute), 68  
`forks_url` (`github.Repository.Repository` attribute), 39  
`full_name` (`github.Repository.Repository` attribute), 39

## G

`get()` (`github.StatsPunchCard.StatsPunchCard` method), 86  
`get_api_status()` (`github.MainClass.Github` method), 7

- `get_api_status_messages()` (github.MainClass.Github method), 7
- `get_archive_link()` (github.Repository.Repository method), 45
- `get_assignees()` (github.Repository.Repository method), 45
- `get_authorization()` (github.AuthenticatedUser.AuthenticatedUser method), 63
- `get_authorizations()` (github.AuthenticatedUser.AuthenticatedUser method), 63
- `get_branch()` (github.Repository.Repository method), 45
- `get_branches()` (github.Repository.Repository method), 45
- `get_collaborators()` (github.Repository.Repository method), 46
- `get_comment()` (github.Gist.Gist method), 84
- `get_comment()` (github.Issue.Issue method), 76
- `get_comment()` (github.PullRequest.PullRequest method), 32
- `get_comment()` (github.Repository.Repository method), 46
- `get_comments()` (github.Commit.Commit method), 36
- `get_comments()` (github.Gist.Gist method), 84
- `get_comments()` (github.Issue.Issue method), 76
- `get_comments()` (github.PullRequest.PullRequest method), 33
- `get_comments()` (github.Repository.Repository method), 46
- `get_commit()` (github.Repository.Repository method), 46
- `get_commits()` (github.PullRequest.PullRequest method), 33
- `get_commits()` (github.Repository.Repository method), 46
- `get_contents()` (github.Repository.Repository method), 46
- `get_contributors()` (github.Repository.Repository method), 47
- `get_dir_contents()` (github.Repository.Repository method), 47
- `get_download()` (github.Repository.Repository method), 47
- `get_downloads()` (github.Repository.Repository method), 47
- `get_emails()` (github.AuthenticatedUser.AuthenticatedUser method), 64
- `get_emojis()` (github.MainClass.Github method), 6
- `get_events()` (github.AuthenticatedUser.AuthenticatedUser method), 64
- `get_events()` (github.Issue.Issue method), 76
- `get_events()` (github.NamedUser.NamedUser method), 72
- `get_events()` (github.Organization.Organization method), 27
- `get_events()` (github.Repository.Repository method), 47
- `get_file_contents()` (github.Repository.Repository method), 46
- `get_files()` (github.PullRequest.PullRequest method), 33
- `get_followers()` (github.AuthenticatedUser.AuthenticatedUser method), 64
- `get_followers()` (github.NamedUser.NamedUser method), 73
- `get_following()` (github.AuthenticatedUser.AuthenticatedUser method), 64
- `get_following()` (github.NamedUser.NamedUser method), 73
- `get_forks()` (github.Repository.Repository method), 47
- `get_gist()` (github.MainClass.Github method), 4
- `get_gists()` (github.AuthenticatedUser.AuthenticatedUser method), 64
- `get_gists()` (github.MainClass.Github method), 4
- `get_gists()` (github.NamedUser.NamedUser method), 73
- `get_git_blob()` (github.Repository.Repository method), 47
- `get_git_commit()` (github.Repository.Repository method), 47
- `get_git_ref()` (github.Repository.Repository method), 48
- `get_git_refs()` (github.Repository.Repository method), 48
- `get_git_tag()` (github.Repository.Repository method), 48
- `get_git_tree()` (github.Repository.Repository method), 48
- `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.Repository.Repository method), 48
- `get_hooks()` (github.MainClass.Github method), 6
- `get_hooks()` (github.Repository.Repository method), 48
- `get_issue()` (github.Repository.Repository method), 48
- `get_issue_comment()` (github.PullRequest.PullRequest method), 33
- `get_issue_comments()` (github.PullRequest.PullRequest method), 33
- `get_issues()` (github.AuthenticatedUser.AuthenticatedUser method), 64
- `get_issues()` (github.Organization.Organization method), 27
- `get_issues()` (github.Repository.Repository method), 48
- `get_issues_comments()` (github.Repository.Repository method), 49
- `get_issues_event()` (github.Repository.Repository method), 49
- `get_issues_events()` (github.Repository.Repository method), 49
- `get_key()` (github.AuthenticatedUser.AuthenticatedUser method), 65
- `get_key()` (github.Repository.Repository method), 49
- `get_keys()` (github.AuthenticatedUser.AuthenticatedUser method), 65
- `get_keys()` (github.NamedUser.NamedUser method), 73
- `get_keys()` (github.Repository.Repository method), 49
- `get_label()` (github.Repository.Repository method), 49



`get_labels()` (`github.Issue.Issue` method), 76  
`get_labels()` (`github.Milestone.Milestone` method), 82  
`get_labels()` (`github.Repository.Repository` method), 49  
`get_languages()` (`github.Repository.Repository` method), 49  
`get_last_api_status_message()` (`github.MainClass.Github` method), 7  
`get_members()` (`github.Organization.Organization` method), 27  
`get_members()` (`github.Team.Team` method), 88  
`get_milestone()` (`github.Repository.Repository` method), 50  
`get_milestones()` (`github.Repository.Repository` method), 50  
`get_network_events()` (`github.Repository.Repository` method), 50  
`get_notification()` (`github.AuthenticatedUser.AuthenticatedUser` method), 65  
`get_notifications()` (`github.AuthenticatedUser.AuthenticatedUser` method), 65  
`get_organization()` (`github.MainClass.Github` method), 4  
`get_organization_events()` (`github.AuthenticatedUser.AuthenticatedUser` method), 65  
`get_orgs()` (`github.AuthenticatedUser.AuthenticatedUser` method), 65  
`get_orgs()` (`github.NamedUser.NamedUser` method), 73  
`get_public_events()` (`github.NamedUser.NamedUser` method), 73  
`get_public_members()` (`github.Organization.Organization` method), 27  
`get_public_received_events()` (`github.NamedUser.NamedUser` method), 73  
`get_pull()` (`github.Repository.Repository` method), 50  
`get_pulls()` (`github.Repository.Repository` method), 50  
`get_pulls_comments()` (`github.Repository.Repository` method), 50  
`get_pulls_review_comments()` (`github.Repository.Repository` method), 50  
`get_rate_limit()` (`github.MainClass.Github` method), 3  
`get_readme()` (`github.Repository.Repository` method), 51  
`get_received_events()` (`github.NamedUser.NamedUser` method), 73  
`get_repo()` (`github.AuthenticatedUser.AuthenticatedUser` method), 65  
`get_repo()` (`github.MainClass.Github` method), 4  
`get_repo()` (`github.NamedUser.NamedUser` method), 73  
`get_repo()` (`github.Organization.Organization` method), 27  
`get_repos()` (`github.AuthenticatedUser.AuthenticatedUser` method), 65  
`get_repos()` (`github.MainClass.Github` method), 4  
`get_repos()` (`github.NamedUser.NamedUser` method), 73  
`get_repos()` (`github.Organization.Organization` method), 28  
`get_repos()` (`github.Team.Team` method), 88  
`get_review_comment()` (`github.PullRequest.PullRequest` method), 33  
`get_review_comments()` (`github.PullRequest.PullRequest` method), 33  
`get_stargazers()` (`github.Repository.Repository` method), 51  
`get_starred()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66  
`get_starred()` (`github.NamedUser.NamedUser` method), 73  
`get_starred_gists()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66  
`get_stats_code_frequency()` (`github.Repository.Repository` method), 51  
`get_stats_commit_activity()` (`github.Repository.Repository` method), 51  
`get_stats_contributors()` (`github.Repository.Repository` method), 51  
`get_stats_participation()` (`github.Repository.Repository` method), 51  
`get_stats_punch_card()` (`github.Repository.Repository` method), 51  
`get_statuses()` (`github.Commit.Commit` method), 36  
`get_subscribers()` (`github.Repository.Repository` method), 51  
`get_subscriptions()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66  
`get_subscriptions()` (`github.NamedUser.NamedUser` method), 74  
`get_tags()` (`github.Repository.Repository` method), 51  
`get_team()` (`github.Organization.Organization` method), 28  
`get_teams()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66  
`get_teams()` (`github.Organization.Organization` method), 28  
`get_teams()` (`github.Repository.Repository` method), 51  
`get_user()` (`github.MainClass.Github` method), 4  
`get_user_issues()` (`github.AuthenticatedUser.AuthenticatedUser` method), 64  
`get_users()` (`github.MainClass.Github` method), 4  
`get_watched()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66  
`get_watched()` (`github.NamedUser.NamedUser` method), 74  
`get_watchers()` (`github.Repository.Repository` method), 52  
`Gist` (class in `github.Gist`), 83  
`GistComment` (class in `github.GistComment`), 29  
`GistFile` (class in `github.GistFile`), 81  
`GistHistoryState` (class in `github.GistHistoryState`), 68



- `gists_url` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60
- `gists_url` (`github.NamedUser.NamedUser` attribute), 71
- `git_commits_url` (`github.Repository.Repository` attribute), 39
- `git_pull_url` (`github.Gist.Gist` attribute), 83
- `git_pull_url` (`github.GistHistoryState.GistHistoryState` attribute), 68
- `git_push_url` (`github.Gist.Gist` attribute), 83
- `git_push_url` (`github.GistHistoryState.GistHistoryState` attribute), 68
- `git_refs_url` (`github.Repository.Repository` attribute), 39
- `git_tags_url` (`github.Repository.Repository` attribute), 39
- `git_url` (`github.ContentFile.ContentFile` attribute), 29
- `git_url` (`github.Repository.Repository` attribute), 40
- `GitAuthor` (class in `github.GitAuthor`), 34
- `GitBlob` (class in `github.GitBlob`), 36
- `GitCommit` (class in `github.GitCommit`), 55
- `Github` (class in `github.MainClass`), 3
- `github` (module), 3
- `github.GithubException` (module), 19
- `GithubException`, 19
- `GithubObject` (class in `github.GithubObject`), 21
- `GitignoreTemplate` (class in `github.GitignoreTemplate`), 88
- `GitObject` (class in `github.GitObject`), 69
- `GitRef` (class in `github.GitRef`), 53
- `GitTag` (class in `github.GitTag`), 80
- `GitTree` (class in `github.GitTree`), 54
- `GitTreeElement` (class in `github.GitTreeElement`), 58
- `gravatar_id` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60
- `gravatar_id` (`github.NamedUser.NamedUser` attribute), 71
- `gravatar_id` (`github.Organization.Organization` attribute), 25
- H**
- `has_downloads` (`github.Repository.Repository` attribute), 40
- `has_in_assignees()` (`github.Repository.Repository` method), 52
- `has_in_collaborators()` (`github.Repository.Repository` method), 52
- `has_in_following()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66
- `has_in_following()` (`github.NamedUser.NamedUser` method), 74
- `has_in_members()` (`github.Organization.Organization` method), 28
- `has_in_members()` (`github.Team.Team` method), 88
- `has_in_public_members()` (`github.Organization.Organization` method), 28
- `has_in_repos()` (`github.Team.Team` method), 88
- `has_in_starred()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66
- `has_in_subscriptions()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66
- `has_in_watched()` (`github.AuthenticatedUser.AuthenticatedUser` method), 66
- `has_issues` (`github.Repository.Repository` attribute), 40
- `has_wiki` (`github.Repository.Repository` attribute), 40
- `head` (`github.PullRequest.PullRequest` attribute), 31
- `hireable` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60
- `hireable` (`github.NamedUser.NamedUser` attribute), 71
- `history` (`github.Gist.Gist` attribute), 83
- `history` (`github.GistHistoryState.GistHistoryState` attribute), 68
- `homepage` (`github.Repository.Repository` attribute), 40
- `Hook` (class in `github.Hook`), 56
- `HookDescription` (class in `github.HookDescription`), 58
- `HookResponse` (class in `github.HookResponse`), 22
- `hooks_url` (`github.Repository.Repository` attribute), 40
- `html_url` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60
- `html_url` (`github.Commit.Commit` attribute), 35
- `html_url` (`github.CommitComment.CommitComment` attribute), 23
- `html_url` (`github.Comparison.Comparison` attribute), 80
- `html_url` (`github.ContentFile.ContentFile` attribute), 29
- `html_url` (`github.Download.Download` attribute), 70
- `html_url` (`github.Gist.Gist` attribute), 83
- `html_url` (`github.GistHistoryState.GistHistoryState` attribute), 68
- `html_url` (`github.GitCommit.GitCommit` attribute), 55
- `html_url` (`github.Issue.Issue` attribute), 75
- `html_url` (`github.IssueComment.IssueComment` attribute), 38
- `html_url` (`github.IssuePullRequest.IssuePullRequest` attribute), 58
- `html_url` (`github.NamedUser.NamedUser` attribute), 71
- `html_url` (`github.Organization.Organization` attribute), 25
- `html_url` (`github.PullRequest.PullRequest` attribute), 31
- `html_url` (`github.PullRequestComment.PullRequestComment` attribute), 56
- `html_url` (`github.Repository.Repository` attribute), 40
- `id` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60
- `id` (`github.Authorization.Authorization` attribute), 77
- `id` (`github.CommitComment.CommitComment` attribute), 23
- `id` (`github.CommitStatus.CommitStatus` attribute), 24
- `id` (`github.Download.Download` attribute), 70
- `id` (`github.Event.Event` attribute), 53
- `id` (`github.Gist.Gist` attribute), 84

`id` (`github.GistComment.GistComment` attribute), 29  
`id` (`github.GistHistoryState.GistHistoryState` attribute), 68  
`id` (`github.Hook.Hook` attribute), 57  
`id` (`github.Issue.Issue` attribute), 75  
`id` (`github.IssueComment.IssueComment` attribute), 38  
`id` (`github.IssueEvent.IssueEvent` attribute), 21  
`id` (`github.Milestone.Milestone` attribute), 82  
`id` (`github.NamedUser.NamedUser` attribute), 71  
`id` (`github.Notification.Notification` attribute), 85  
`id` (`github.Organization.Organization` attribute), 25  
`id` (`github.PullRequest.PullRequest` attribute), 31  
`id` (`github.PullRequestComment.PullRequestComment` attribute), 56  
`id` (`github.Repository.Repository` attribute), 40  
`id` (`github.RepositoryKey.RepositoryKey` attribute), 59  
`id` (`github.Team.Team` attribute), 87  
`id` (`github.UserKey.UserKey` attribute), 79  
`InputFileContent` (class in `github.InputFileContent`), 20  
`InputGitAuthor` (class in `github.InputGitAuthor`), 20  
`InputGitTreeElement` (class in `github.InputGitTreeElement`), 20  
`is_merged()` (`github.PullRequest.PullRequest` method), 33  
`is_starred()` (`github.Gist.Gist` method), 85  
`Issue` (class in `github.Issue`), 74  
`issue` (`github.IssueEvent.IssueEvent` attribute), 21  
`issue_comment_url` (`github.Repository.Repository` attribute), 40  
`issue_events_url` (`github.Repository.Repository` attribute), 40  
`issue_url` (`github.IssueComment.IssueComment` attribute), 38  
`issue_url` (`github.PullRequest.PullRequest` attribute), 31  
`IssueComment` (class in `github.IssueComment`), 37  
`IssueEvent` (class in `github.IssueEvent`), 21  
`IssuePullRequest` (class in `github.IssuePullRequest`), 57  
`issues_url` (`github.Repository.Repository` attribute), 40

## K

`key` (`github.RepositoryKey.RepositoryKey` attribute), 59  
`key` (`github.UserKey.UserKey` attribute), 79  
`keys_url` (`github.Repository.Repository` attribute), 40

## L

`Label` (class in `github.Label`), 35  
`label` (`github.PullRequestPart.PullRequestPart` attribute), 22  
`labels` (`github.Issue.Issue` attribute), 75  
`labels_url` (`github.Issue.Issue` attribute), 75  
`labels_url` (`github.Milestone.Milestone` attribute), 82  
`labels_url` (`github.Repository.Repository` attribute), 40  
`language` (`github.GistFile.GistFile` attribute), 81  
`language` (`github.Repository.Repository` attribute), 40

`languages_url` (`github.Repository.Repository` attribute), 40  
`last_modified` (`github.GithubObject.GithubObject` attribute), 21  
`last_read_at` (`github.Notification.Notification` attribute), 85  
`last_response` (`github.Hook.Hook` attribute), 57  
`last_updated` (`github.Status.Status` attribute), 83  
`latest_comment_url` (`github.NotificationSubject.NotificationSubject` attribute), 34  
`legacy_search_issues()` (`github.Repository.Repository` method), 52  
`legacy_search_repos()` (`github.MainClass.Github` method), 4  
`legacy_search_user_by_email()` (`github.MainClass.Github` method), 5  
`legacy_search_users()` (`github.MainClass.Github` method), 5  
`limit` (`github.Rate.Rate` attribute), 78  
`line` (`github.CommitComment.CommitComment` attribute), 23  
`load()` (`github.MainClass.Github` method), 7  
`location` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`location` (`github.NamedUser.NamedUser` attribute), 71  
`location` (`github.Organization.Organization` attribute), 25  
`login` (`github.AuthenticatedUser.AuthenticatedUser` attribute), 60  
`login` (`github.NamedUser.NamedUser` attribute), 72  
`login` (`github.Organization.Organization` attribute), 25

## M

`master_branch` (`github.Repository.Repository` attribute), 40  
`members_count` (`github.Team.Team` attribute), 87  
`members_url` (`github.Organization.Organization` attribute), 25  
`members_url` (`github.Team.Team` attribute), 87  
`merge()` (`github.PullRequest.PullRequest` method), 33  
`merge()` (`github.Repository.Repository` method), 52  
`merge_base_commit` (`github.Comparison.Comparison` attribute), 80  
`merge_commit_sha` (`github.PullRequest.PullRequest` attribute), 31  
`mergeable` (`github.PullRequest.PullRequest` attribute), 31  
`mergeable_state` (`github.PullRequest.PullRequest` attribute), 31  
`merged` (`github.PullRequest.PullRequest` attribute), 31  
`merged` (`github.PullRequestMergeStatus.PullRequestMergeStatus` attribute), 86  
`merged_at` (`github.PullRequest.PullRequest` attribute), 31  
`merged_by` (`github.PullRequest.PullRequest` attribute), 31  
`merges_url` (`github.Repository.Repository` attribute), 40  
`message` (`github.GitCommit.GitCommit` attribute), 55

- message (github.GitTag.GitTag attribute), 80
- message (github.HookResponse.HookResponse attribute), 22
- message (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 86
- Milestone (class in github.Milestone), 81
- milestone (github.Issue.Issue attribute), 75
- milestone (github.PullRequest.PullRequest attribute), 31
- milestones\_url (github.Repository.Repository attribute), 40
- mime\_type (github.Download.Download attribute), 70
- mirror\_url (github.Repository.Repository attribute), 41
- mode (github.GitTreeElement.GitTreeElement attribute), 58
- ## N
- name (github.AuthenticatedUser.AuthenticatedUser attribute), 60
- name (github.AuthorizationApplication.AuthorizationApplication attribute), 54
- name (github.Branch.Branch attribute), 78
- name (github.ContentFile.ContentFile attribute), 29
- name (github.Download.Download attribute), 70
- name (github.GitAuthor.GitAuthor attribute), 34
- name (github.GitignoreTemplate.GitignoreTemplate attribute), 88
- name (github.Hook.Hook attribute), 57
- name (github.HookDescription.HookDescription attribute), 58
- name (github.Label.Label attribute), 35
- name (github.NamedUser.NamedUser attribute), 72
- name (github.Organization.Organization attribute), 25
- name (github.Plan.Plan attribute), 34
- name (github.Repository.Repository attribute), 41
- name (github.Tag.Tag attribute), 86
- name (github.Team.Team attribute), 87
- NamedUser (class in github.NamedUser), 70
- network\_count (github.Repository.Repository attribute), 41
- note (github.Authorization.Authorization attribute), 77
- note\_url (github.Authorization.Authorization attribute), 77
- Notification (class in github.Notification), 85
- notifications\_url (github.Repository.Repository attribute), 41
- NotificationSubject (class in github.NotificationSubject), 34
- number (github.Issue.Issue attribute), 75
- number (github.Milestone.Milestone attribute), 82
- number (github.PullRequest.PullRequest attribute), 31
- ## O
- oauth\_scopes (github.MainClass.Github attribute), 4
- object (github.GitRef.GitRef attribute), 53
- object (github.GitTag.GitTag attribute), 80
- open\_issues (github.Milestone.Milestone attribute), 82
- open\_issues (github.Repository.Repository attribute), 41
- open\_issues\_count (github.Repository.Repository attribute), 41
- org (github.Event.Event attribute), 53
- Organization (class in github.Organization), 24
- organization (github.Repository.Repository attribute), 41
- organizations\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 60
- organizations\_url (github.NamedUser.NamedUser attribute), 72
- original\_commit\_id (github.PullRequestComment.PullRequestComment attribute), 56
- original\_position (github.PullRequestComment.PullRequestComment attribute), 56
- owned\_private\_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- owned\_private\_repos (github.NamedUser.NamedUser attribute), 72
- owned\_private\_repos (github.Organization.Organization attribute), 25
- owner (github.Gist.Gist attribute), 84
- owner (github.GistHistoryState.GistHistoryState attribute), 68
- owner (github.Repository.Repository attribute), 41
- owner (github.StatsParticipation.StatsParticipation attribute), 67
- ## P
- PaginatedList (class in github.PaginatedList), 20
- parent (github.Repository.Repository attribute), 41
- parents (github.Commit.Commit attribute), 35
- parents (github.GitCommit.GitCommit attribute), 55
- patch (github.File.File attribute), 37
- patch\_url (github.Comparison.Comparison attribute), 80
- patch\_url (github.IssuePullRequest.IssuePullRequest attribute), 58
- patch\_url (github.PullRequest.PullRequest attribute), 31
- path (github.CommitComment.CommitComment attribute), 23
- path (github.ContentFile.ContentFile attribute), 29
- path (github.Download.Download attribute), 70
- path (github.GitTreeElement.GitTreeElement attribute), 58
- path (github.PullRequestComment.PullRequestComment attribute), 56
- payload (github.Event.Event attribute), 53
- per\_page (github.MainClass.Github attribute), 3
- permalink\_url (github.Comparison.Comparison attribute), 81
- permission (github.Team.Team attribute), 87
- Permissions (class in github.Permissions), 74
- permissions (github.Repository.Repository attribute), 41

- Plan (class in github.Plan), 34
- plan (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- plan (github.NamedUser.NamedUser attribute), 72
- plan (github.Organization.Organization attribute), 25
- policy (github.Download.Download attribute), 70
- position (github.CommitComment.CommitComment attribute), 23
- position (github.PullRequestComment.PullRequestComment attribute), 56
- prefix (github.Download.Download attribute), 70
- private (github.Repository.Repository attribute), 41
- private\_gists (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- private\_gists (github.NamedUser.NamedUser attribute), 72
- private\_gists (github.Organization.Organization attribute), 25
- private\_repos (github.Plan.Plan attribute), 34
- public (github.Event.Event attribute), 53
- public (github.Gist.Gist attribute), 84
- public (github.GistHistoryState.GistHistoryState attribute), 68
- public\_gists (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- public\_gists (github.NamedUser.NamedUser attribute), 72
- public\_gists (github.Organization.Organization attribute), 25
- public\_members\_url (github.Organization.Organization attribute), 25
- public\_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- public\_repos (github.NamedUser.NamedUser attribute), 72
- public\_repos (github.Organization.Organization attribute), 25
- pull (github.Permissions.Permissions attribute), 74
- pull\_request (github.Issue.Issue attribute), 75
- pull\_request\_url (github.PullRequestComment.PullRequestComment attribute), 56
- PullRequest (class in github.PullRequest), 30
- PullRequestComment (class in github.PullRequestComment), 55
- PullRequestMergeStatus (class in github.PullRequestMergeStatus), 86
- PullRequestPart (class in github.PullRequestPart), 22
- pulls\_url (github.Repository.Repository attribute), 41
- push (github.Permissions.Permissions attribute), 74
- pushed\_at (github.Repository.Repository attribute), 41
- rate\_limiting (github.MainClass.Github attribute), 3
- rate\_limiting\_resettime (github.MainClass.Github attribute), 3
- RateLimit (class in github.RateLimit), 58
- RateLimitExceededException, 19
- raw\_data (github.GithubObject.GithubObject attribute), 21
- raw\_headers (github.GithubObject.GithubObject attribute), 21
- raw\_url (github.File.File attribute), 37
- raw\_url (github.GistFile.GistFile attribute), 81
- reason (github.Notification.Notification attribute), 85
- received\_events\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- received\_events\_url (github.NamedUser.NamedUser attribute), 72
- redirect (github.Download.Download attribute), 70
- ref (github.GitRef.GitRef attribute), 53
- ref (github.PullRequestPart.PullRequestPart attribute), 22
- remaining (github.Rate.Rate attribute), 78
- remove\_from\_collaborators() (github.Repository.Repository method), 52
- remove\_from\_emails() (github.AuthenticatedUser.AuthenticatedUser method), 67
- remove\_from\_following() (github.AuthenticatedUser.AuthenticatedUser method), 67
- remove\_from\_labels() (github.Issue.Issue method), 76
- remove\_from\_members() (github.Organization.Organization method), 28
- remove\_from\_members() (github.Team.Team method), 88
- remove\_from\_public\_members() (github.Organization.Organization method), 28
- remove\_from\_repos() (github.Team.Team method), 88
- remove\_from\_starred() (github.AuthenticatedUser.AuthenticatedUser method), 67
- remove\_from\_subscriptions() (github.AuthenticatedUser.AuthenticatedUser method), 67
- remove\_from\_watched() (github.AuthenticatedUser.AuthenticatedUser method), 67
- in render\_markdown() (github.MainClass.Github method), 6
- in repo (github.Event.Event attribute), 53
- repo (github.PullRequestPart.PullRequestPart attribute), 22
- repos\_count (github.Team.Team attribute), 87
- repos\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- repos\_url (github.NamedUser.NamedUser attribute), 72
- repos\_url (github.Organization.Organization attribute), 25
- repositories\_url (github.Team.Team attribute), 87

## R

- Rate (class in github.Rate), 78
- rate (github.RateLimit.RateLimit attribute), 58

Repository (class in github.Repository), 38  
 repository (github.ContentFile.ContentFile attribute), 29  
 repository (github.Issue.Issue attribute), 75  
 repository (github.Notification.Notification attribute), 85  
 RepositoryKey (class in github.RepositoryKey), 59  
 reset (github.Rate.Rate attribute), 78  
 reset\_starred() (github.Gist.Gist method), 85  
 review\_comment\_url (github.PullRequest.PullRequest attribute), 31  
 review\_comments (github.PullRequest.PullRequest attribute), 31  
 review\_comments\_url (github.PullRequest.PullRequest attribute), 31

## S

s3\_url (github.Download.Download attribute), 70  
 schema (github.HookDescription.HookDescription attribute), 58  
 scopes (github.Authorization.Authorization attribute), 77  
 search\_code() (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), 77  
 set\_starred() (github.Gist.Gist method), 85  
 sha (github.Commit.Commit attribute), 35  
 sha (github.ContentFile.ContentFile attribute), 29  
 sha (github.File.File attribute), 37  
 sha (github.GitBlob.GitBlob attribute), 36  
 sha (github.GitCommit.GitCommit attribute), 55  
 sha (github.GitObject.GitObject attribute), 69  
 sha (github.GitTag.GitTag attribute), 80  
 sha (github.GitTree.GitTree attribute), 54  
 sha (github.GitTreeElement.GitTreeElement attribute), 58  
 sha (github.PullRequestMergeStatus.PullRequestMergeStatus attribute), 86  
 sha (github.PullRequestPart.PullRequestPart attribute), 22  
 signature (github.Download.Download attribute), 70  
 site\_admin (github.AuthenticatedUser.AuthenticatedUser attribute), 61  
 size (github.ContentFile.ContentFile attribute), 29  
 size (github.Download.Download attribute), 70  
 size (github.GistFile.GistFile attribute), 81  
 size (github.GitBlob.GitBlob attribute), 37  
 size (github.GitTreeElement.GitTreeElement attribute), 58  
 size (github.Repository.Repository attribute), 41  
 slug (github.Team.Team attribute), 87  
 source (github.GitignoreTemplate.GitignoreTemplate attribute), 88  
 source (github.Repository.Repository attribute), 41  
 space (github.Plan.Plan attribute), 34  
 ssh\_url (github.Repository.Repository attribute), 41  
 stargazers\_count (github.Repository.Repository attribute), 41  
 stargazers\_url (github.Repository.Repository attribute), 41  
 starred\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 61  
 starred\_url (github.NamedUser.NamedUser attribute), 72  
 state (github.CommitStatus.CommitStatus attribute), 24  
 state (github.Issue.Issue attribute), 75  
 state (github.Milestone.Milestone attribute), 82  
 state (github.PullRequest.PullRequest attribute), 31  
 stats (github.Commit.Commit attribute), 36  
 StatsCodeFrequency (class in github.StatsCodeFrequency), 86  
 StatsCommitActivity (class in github.StatsCommitActivity), 79  
 StatsContributor (class in github.StatsContributor), 21  
 StatsContributor.Week (class in github.StatsContributor), 21  
 StatsParticipation (class in github.StatsParticipation), 67  
 StatsPunchCard (class in github.StatsPunchCard), 86  
 Status (class in github.Status), 83  
 status (github.Comparison.Comparison attribute), 81  
 status (github.File.File attribute), 37  
 status (github.GithubException.GithubException attribute), 19  
 status (github.HookResponse.HookResponse attribute), 22  
 status (github.Status.Status attribute), 83  
 status (github.StatusMessage.StatusMessage attribute), 77  
 statuses\_url (github.Repository.Repository attribute), 42  
 StatusMessage (class in github.StatusMessage), 77  
 subject (github.Notification.Notification attribute), 85  
 subscribe\_to\_hub() (github.Repository.Repository method), 52  
 subscribers\_url (github.Repository.Repository attribute), 42  
 subscription\_url (github.Notification.Notification attribute), 85  
 subscription\_url (github.Repository.Repository attribute), 42  
 subscriptions\_url (github.AuthenticatedUser.AuthenticatedUser attribute), 61  
 subscriptions\_url (github.NamedUser.NamedUser attribute), 72  
 supported\_events (github.HookDescription.HookDescription attribute), 58  
 svn\_url (github.Repository.Repository attribute), 42

## T

Tag (class in github.Tag), 86  
 tag (github.GitTag.GitTag attribute), 80



- tagger (github.GitTag.GitTag attribute), 80
- tags\_url (github.Repository.Repository attribute), 42
- tarball\_url (github.Tag.Tag attribute), 86
- target\_url (github.CommitStatus.CommitStatus attribute), 24
- Team (class in github.Team), 87
- teams\_url (github.Repository.Repository attribute), 42
- test() (github.Hook.Hook method), 57
- test\_url (github.Hook.Hook attribute), 57
- title (github.Issue.Issue attribute), 75
- title (github.Milestone.Milestone attribute), 82
- title (github.NotificationSubject.NotificationSubject attribute), 34
- title (github.PullRequest.PullRequest attribute), 31
- title (github.RepositoryKey.RepositoryKey attribute), 59
- title (github.UserKey.UserKey attribute), 79
- token (github.Authorization.Authorization attribute), 77
- total (github.CommitStats.CommitStats attribute), 54
- total (github.StatsCommitActivity.StatsCommitActivity attribute), 79
- total (github.StatsContributor.StatsContributor attribute), 22
- total\_commits (github.Comparison.Comparison attribute), 81
- total\_private\_repos (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- total\_private\_repos (github.NamedUser.NamedUser attribute), 72
- total\_private\_repos (github.Organization.Organization attribute), 26
- transformation\_exception (github.GithubException.BadAttributeException attribute), 19
- tree (github.GitCommit.GitCommit attribute), 55
- tree (github.GitTree.GitTree attribute), 54
- trees\_url (github.Repository.Repository attribute), 42
- TwoFactorException, 19
- type (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- type (github.ContentFile.ContentFile attribute), 29
- type (github.Event.Event attribute), 53
- type (github.GistFile.GistFile attribute), 81
- type (github.GitObject.GitObject attribute), 69
- type (github.GitTreeElement.GitTreeElement attribute), 58
- type (github.NamedUser.NamedUser attribute), 72
- type (github.NotificationSubject.NotificationSubject attribute), 34
- type (github.Organization.Organization attribute), 26
- unsubscribe\_from\_hub() (github.Repository.Repository method), 53
- updated\_at (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- updated\_at (github.Authorization.Authorization attribute), 77
- updated\_at (github.CommitComment.CommitComment attribute), 23
- updated\_at (github.CommitStatus.CommitStatus attribute), 24
- updated\_at (github.Gist.Gist attribute), 84
- updated\_at (github.GistComment.GistComment attribute), 29
- updated\_at (github.GistHistoryState.GistHistoryState attribute), 69
- updated\_at (github.Hook.Hook attribute), 57
- updated\_at (github.Issue.Issue attribute), 75
- updated\_at (github.IssueComment.IssueComment attribute), 38
- updated\_at (github.Milestone.Milestone attribute), 82
- updated\_at (github.NamedUser.NamedUser attribute), 72
- updated\_at (github.Notification.Notification attribute), 85
- updated\_at (github.Organization.Organization attribute), 26
- updated\_at (github.PullRequest.PullRequest attribute), 32
- updated\_at (github.PullRequestComment.PullRequestComment attribute), 56
- updated\_at (github.Repository.Repository attribute), 42
- url (github.AuthenticatedUser.AuthenticatedUser attribute), 61
- url (github.Authorization.Authorization attribute), 77
- url (github.AuthorizationApplication.AuthorizationApplication attribute), 54
- url (github.Commit.Commit attribute), 36
- url (github.CommitComment.CommitComment attribute), 23
- url (github.CommitStatus.CommitStatus attribute), 24
- url (github.Comparison.Comparison attribute), 81
- url (github.ContentFile.ContentFile attribute), 29
- url (github.Download.Download attribute), 70
- url (github.Gist.Gist attribute), 84
- url (github.GistComment.GistComment attribute), 29
- url (github.GistHistoryState.GistHistoryState attribute), 69
- url (github.GitBlob.GitBlob attribute), 37
- url (github.GitCommit.GitCommit attribute), 55
- url (github.GitObject.GitObject attribute), 69
- url (github.GitRef.GitRef attribute), 54
- url (github.GitTag.GitTag attribute), 80
- url (github.GitTree.GitTree attribute), 55
- url (github.GitTreeElement.GitTreeElement attribute), 59
- url (github.Hook.Hook attribute), 57
- url (github.Issue.Issue attribute), 75
- url (github.IssueComment.IssueComment attribute), 38

## U

- UnknownObjectException, 19
- unread (github.Notification.Notification attribute), 85

[url \(github.IssueEvent.IssueEvent attribute\), 21](#)  
[url \(github.Label.Label attribute\), 35](#)  
[url \(github.Milestone.Milestone attribute\), 82](#)  
[url \(github.NamedUser.NamedUser attribute\), 72](#)  
[url \(github.Notification.Notification attribute\), 85](#)  
[url \(github.NotificationSubject.NotificationSubject attribute\), 34](#)  
[url \(github.Organization.Organization attribute\), 26](#)  
[url \(github.PullRequest.PullRequest attribute\), 32](#)  
[url \(github.PullRequestComment.PullRequestComment attribute\), 56](#)  
[url \(github.Repository.Repository attribute\), 42](#)  
[url \(github.RepositoryKey.RepositoryKey attribute\), 59](#)  
[url \(github.Team.Team attribute\), 87](#)  
[url \(github.UserKey.UserKey attribute\), 79](#)  
[user \(github.CommitComment.CommitComment attribute\), 23](#)  
[user \(github.Gist.Gist attribute\), 84](#)  
[user \(github.GistComment.GistComment attribute\), 29](#)  
[user \(github.GistHistoryState.GistHistoryState attribute\), 69](#)  
[user \(github.Issue.Issue attribute\), 75](#)  
[user \(github.IssueComment.IssueComment attribute\), 38](#)  
[user \(github.PullRequest.PullRequest attribute\), 32](#)  
[user \(github.PullRequestComment.PullRequestComment attribute\), 56](#)  
[user \(github.PullRequestPart.PullRequestPart attribute\), 22](#)  
[UserKey \(class in github.UserKey\), 79](#)

## V

[verified \(github.RepositoryKey.RepositoryKey attribute\), 59](#)  
[verified \(github.UserKey.UserKey attribute\), 79](#)  
[version \(github.GistHistoryState.GistHistoryState attribute\), 69](#)

## W

[w \(github.StatsContributor.StatsContributor.Week attribute\), 21](#)  
[watchers \(github.Repository.Repository attribute\), 42](#)  
[watchers\\_count \(github.Repository.Repository attribute\), 42](#)  
[week \(github.StatsCodeFrequency.StatsCodeFrequency attribute\), 86](#)  
[week \(github.StatsCommitActivity.StatsCommitActivity attribute\), 79](#)  
[weeks \(github.StatsContributor.StatsContributor attribute\), 22](#)

## Z

[zipball\\_url \(github.Tag.Tag attribute\), 86](#)