
aiogithub Documentation

Release 0.1.dev81+g03913c6

Reupen Shah

Jun 26, 2018

1	Installation	3
1.1	Prerequisites	3
1.2	Installation of development versions	3
2	Usage	5
2.1	Development status	5
2.2	A note about asyncio	5
2.3	A simple example	5
2.4	The GitHub class	6
2.5	Authentication	6
3	Client	7
4	Response classes	9
5	User classes	11
6	Organization classes	15
7	Repo classes	17
8	Pull request classes	21
9	Issue classes	23
10	Indexes and tables	25
	Python Module Index	27

Contents:

1.1 Prerequisites

aiogithub requires Python 3.5 or newer.

1.2 Installation of development versions

Currently, only development versions are available. You can install the current development version by running:

```
pip3.5 install -U setuptools  
pip3.5 install git+https://github.com/reupen/aiogithub.git#egg=aiogithub
```


2.1 Development status

This library is a work in progress. So far, select read operations have been implemented.

2.2 A note about asyncio

aiogithub is written on top of asyncio and aiohttp. You should have a basic grasp of asynchronous programming using asyncio in Python 3 before using this library.

In particular, using of the library generally occurs in an `async def` function. Calls to coroutine functions would normally be preceded with `await`, and `async for` would normally be used with asynchronous iterables.

2.3 A simple example

```
import asyncio

from aiogithub import GitHub

async def main():
    async with GitHub() as api:
        user = await api.get_user('reupen')
        print(user.login)
        # user is also a dict, so you can see the underlying data via print(user)

        # Get this user's repos. There's no need to worry about
        # pagination - you can simply iterate over list objects like
        # this and pages will be retrieved as needed:
        async for repo in user.get_repos():
            pass # Do something with each repo here
```

(continues on next page)

(continued from previous page)

```
# Or you can fetch the entire list like this:
repos = await user.get_repos().all()
# Do something with repos

asyncio.get_event_loop().run_until_complete(main())
```

2.4 The GitHub class

The *GitHub* class is the main entry point into aiogithub functionality. It is a wrapper around an aiohttp ClientSession object, which means that you automatically benefit from connection pooling. You should use the object as a context manager to make sure the session is closed as soon as you are done with it.

2.5 Authentication

You can make a limited number of requests to the GitHub API unauthenticated, but for most purposes you will want to be authenticated. To authenticate with GitHub using aiogithub, you should use a personal access token. You can generate a personal access token in [your GitHub settings](#).

aiogithub will use the value of the the GITHUB_TOKEN environment variable to authenticate with GitHub if it is set. You can also pass a token to *GitHub* (which will override any value in the GITHUB_TOKEN environment variable)::

```
with GitHub(token='a_personal_access_token') as api:
    pass
```

```
class aiogithub.GitHub (token: str = None, items_per_page=100, timeout_secs=10,  
                        max_paginated_items=1000, enable_preview_api=False)
```

Initialises a GitHub API client.

If no personal access token is provided, the client will check the GITHUB_TOKEN environment variable for a token and use that if present. If still without a token, the GitHub API will be used unauthenticated.

Parameters

- **token** – GitHub personal access token
- **items_per_page** – Items to request per page, must be between 1 and 100
- **timeout_secs** – Timeout in seconds for HTTP requests
- **max_paginated_items** – Safety limit for when iterating through list results to avoid inadvertently making a huge number of requests

```
coroutine close () → None
```

```
coroutine get_branch (owner_name, repo_name, branch_name) → aiogithub.objects.branch.Branch
```

Gets a single branch of a repository.

```
coroutine get_current_user () → aiogithub.objects.user.AuthenticatedUser
```

Gets the current authenticated user.

```
coroutine get_issue (owner_name, repo_name, issue_number) → aiogithub.objects.issue.Issue
```

Gets a single issue of a repository.

```
coroutine get_organization (username, defer_fetch=False) → aiogithub.objects.organization.Organization
```

Gets a single organization.

Parameters

- **username** – The username/login of the organization to fetch the details of.
- **defer_fetch** – Whether to defer fetching of data about this organization.

Returns An object representing the organization.

coroutine get_pull_request (*owner_name*, *repo_name*, *issue_number*) → aiogithub.objects.pull_request.PullRequest

Gets a single pull request of a repository.

coroutine get_rate_limit () → aiogithub.objects.rate_limit.RateLimit

Gets the current rate limit values.

coroutine get_repo (*owner_name*, *repo_name*, *defer_fetch=False*) → aiogithub.objects.repo.Repo

Gets a single repository.

Parameters

- **owner_name** – The name of the user or organisation that owns the repository.
- **repo_name** – The name of the repository.
- **defer_fetch** – Whether to defer fetching of data about this repository.

Returns An object representing the repository.

get_repos (*since=None*) → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.repo.Repo]

Gets all repos.

coroutine get_user (*username*, *defer_fetch=False*) → aiogithub.objects.user.User

Gets a single user.

Parameters

- **username** – The name of the user to fetch the details of.
- **defer_fetch** – Whether to defer fetching of data about this user.

Returns An object representing the user.

get_users (*since=None*) → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]

Gets all users.

last_rate_limit

The rate limits that were sent by GitHub in the most recent request.

Type Optional[dict]

Response classes

class aiogithub.objects.**BaseObject** (*document*)

Bases: dict

Base class for all data objects

class aiogithub.objects.**BaseResponseObject** (*client*, *document=None*, *limits=None*,
links=None, *fetch_params=None*)

Bases: aiogithub.objects.response.BaseObject

Base class for objects where the contained data corresponds to a response payload of a particular GitHub API URL

coroutine **fetch_data** ()

limits

class aiogithub.objects.**PaginatedListProxy** (*client*, *url*, *element_type*, *fetch_params*)

Bases: typing.AsyncIterable

Public interface to paginated objects

coroutine **all** ()

Returns all items in the collection. This will fetch all result pages that haven't already been fetched.

Use *limit* () when listing large data sets (e.g. all public users) to avoid making a large number of HTTP requests and exhausting your API limits.

limit (*max_items*)

Limits the number of items returned by *all* () or when iterating through the collection elements (using *async for*).

Use when listing large data sets (e.g. all public users) to avoid making a large number of HTTP requests and exhausting your API limits.


```
class aiogithub.objects.PartialUser (client, document=None, limits=None, links=None,  
                                     fetch_params=None)  
    Bases: aiogithub.objects.response.BaseResponseObject  
  
    get_events () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.event.Event]  
    get_followers () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]  
    get_following () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]  
    get_received_events () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.event.Event]  
    get_repos () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.repo.Repo]  
    get_starred () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.repo.Repo]  
    get_subscriptions () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.repo.Repo]  
  
    gravatar_id  
        Type Optional[str]  
  
    html_url  
        Type Optional[str]  
  
    id  
        Type Optional[int]  
  
    login  
        Type str  
  
    site_admin  
        Type Optional[str]  
  
    type  
        Type Optional[str]
```

```

class aiogithub.objects.User (client, document=None, limits=None, links=None,
                                fetch_params=None)
    Bases: aiogithub.objects.user.PartialUser

    bio
        Type Optional[str]

    blog
        Type Optional[str]

    company
        Type Optional[str]

    created_at
        Type Optional[datetime]

    email
        Type Optional[str]

    followers
        Type Optional[int]

    following
        Type Optional[int]

    hireable
        Type Optional[bool]

    location
        Type Optional[str]

    public_gists
        Type Optional[int]

    public_repos
        Type Optional[int]

    updated_at
        Type Optional[datetime]

class aiogithub.objects.AuthenticatedUser (client, document=None, limits=None,
                                              links=None, fetch_params=None)
    Bases: aiogithub.objects.user.User

    collaborators
        Type int

    disk_usage
        Type int

    owned_private_repos
        Type int

    plan
        Type dict
    
```


`private_gists`

Type int

`total_private_repos`

Type int

Organization classes

```
class aiogithub.objects.PartialOrganization(client, document=None, limits=None,  
                                             links=None, fetch_params=None)  
    Bases: aiogithub.objects.response.BaseResponseObject  
  
    avatar_url  
  
    description  
  
    get_events () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.event.Event]  
    get_followers () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]  
    get_following () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]  
    get_gists () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.gist.Gist]  
    get_hooks () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.response.BaseObject]  
    get_issues () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.issue.Issue]  
    get_members () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.PartialUser]  
    get_public_members () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.PartialUser]  
    get_repos () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.repo.Repo]  
  
    id  
  
    login  
  
class aiogithub.objects.Organization(client, document=None, limits=None, links=None,  
                                       fetch_params=None)  
    Bases: aiogithub.objects.organization.PartialOrganization  
  
    blog  
  
    company  
  
    created_at  
  
    email
```

followers
following
html_url
location
name
public_gists
public_repos
type
updated_at

```
class aiogithub.objects.PartialRepo (client, document=None, limits=None, links=None,  
                                     fetch_params=None)  
    Bases: aiogithub.objects.response.BaseResponseObject  
  
    clone_url  
  
    description  
  
    fork  
  
    full_name  
  
    get_assignees () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]  
    get_blobs () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.response.BaseResponseObject]  
    coroutine get_branch (branch) → aiogithub.objects.branch.Branch  
    get_branches () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.branch.Branch]  
    get_collaborators () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]  
    get_comments () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.comment.Comment]  
    get_commits () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.commit.Commit]  
    get_contributors () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.repo.Repo]  
    get_events () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.event.Event]  
    get_forks () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.repo.Repo]  
    get_issues () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.issue.Issue]  
    get_pull_requests () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.pull_request.PullRequest]  
    get_stargazers () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.User]  
  
    git_url  
  
    homepage
```

```
html_url
id
mirror_url
name
owner
private
ssh_url
svn_url

class aiogithub.objects.Repo(client, document=None, limits=None, links=None,
                             fetch_params=None)
    Bases: aiogithub.objects.repo.PartialRepo
    created_at
    default_branch
    forks_count
    has_downloads
    has_issues
    has_pages
    has_wiki
    language
    open_issues_count
    permissions
    pushed_at
    size
    stargazers_count
    updated_at
    watchers_count

class aiogithub.objects.Branch(client, document=None, limits=None, links=None,
                               fetch_params=None)
    Bases: aiogithub.objects.response.BaseResponseObject
    commit
    links
    name

class aiogithub.objects.Commit(client, document=None, limits=None, links=None,
                               fetch_params=None)
    Bases: aiogithub.objects.response.BaseResponseObject
    author
    commit
    committer
```

files

get_comments () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.comment.Comment]

html_url

parents

sha

stats

Pull request classes

```
class aiogithub.objects.PullRequest (client, document=None, limits=None, links=None,  
                                     fetch_params=None)  
    Bases: aiogithub.objects.response.BaseResponseObject  
  
    assignee  
    assignees  
    base  
    body  
    closed_at  
    created_at  
    diff_url  
  
    get_comments () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.comment.Comment]  
    get_commits () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.commit.Commit]  
    coroutine get_issue () → aiogithub.objects.issue.Issue  
    get_requested_reviewers () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.user.PartialUser]  
    coroutine get_review_comment () → aiogithub.objects.review_comment.ReviewComment  
    get_review_comments () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.review_comment.ReviewComment]  
    get_reviews () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.review.Review]  
  
    head  
    html_url  
    id  
    locked  
    merged_at
```

milestone
number
patch_url
requested_reviewers
state
title
updated_at
user

```
class aiogithub.objects.ReviewComment (client, document=None, limits=None, links=None,  
                                         fetch_params=None)
```

```
    Bases: aiogithub.objects.response.BaseResponseObject
```

body
commit_id
created_at
diff_hunk
coroutine **get_pull_request** () → aiogithub.objects.pull_request.PullRequest
html_url
id
original_commit_id
original_position
path
position
updated_at
user

CHAPTER 9

Issue classes

```
class aiogithub.objects.Issue (client, document=None, limits=None, links=None,  
                                fetch_params=None)  
    Bases: aiogithub.objects.response.BaseResponseObject  
  
    assignee  
    assignees  
    body  
    closed_at  
    comments  
    created_at  
    get_comments () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.comment.Comment]  
    get_events () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.event.IssueEvent]  
    get_labels () → aiogithub.objects.response.PaginatedListProxy[aiogithub.objects.response.BaseResponseObject]  
    coroutine get_repo () → aiogithub.objects.repo.Repo  
    html_url  
    id  
    labels  
    locked  
    milestone  
    number  
    state  
    title  
    updated_at
```

user

```
class aiogithub.objects.IssueEvent (client, document=None, limits=None, links=None,  
                                     fetch_params=None)
```

Bases: aiogithub.objects.response.BaseResponseObject

actor

commit_id

created_at

event

id

issue

```
class aiogithub.objects.Comment (client, document=None, limits=None, links=None,  
                                   fetch_params=None)
```

Bases: aiogithub.objects.response.BaseResponseObject

body

created_at

html_url

id

updated_at

user

CHAPTER 10

Indexes and tables

- modindex

a

aiogithub, 7

A

actor (aiogithub.objects.IssueEvent attribute), 24
 aiogithub (module), 7
 all() (aiogithub.objects.PaginatedListProxy method), 9
 assignee (aiogithub.objects.Issue attribute), 23
 assignee (aiogithub.objects.PullRequest attribute), 21
 assignees (aiogithub.objects.Issue attribute), 23
 assignees (aiogithub.objects.PullRequest attribute), 21
 AuthenticatedUser (class in aiogithub.objects), 12
 author (aiogithub.objects.Commit attribute), 18
 avatar_url (aiogithub.objects.PartialOrganization attribute), 15

B

base (aiogithub.objects.PullRequest attribute), 21
 BaseObject (class in aiogithub.objects), 9
 BaseResponseObject (class in aiogithub.objects), 9
 bio (aiogithub.objects.User attribute), 12
 blog (aiogithub.objects.Organization attribute), 15
 blog (aiogithub.objects.User attribute), 12
 body (aiogithub.objects.Comment attribute), 24
 body (aiogithub.objects.Issue attribute), 23
 body (aiogithub.objects.PullRequest attribute), 21
 body (aiogithub.objects.ReviewComment attribute), 22
 Branch (class in aiogithub.objects), 18

C

clone_url (aiogithub.objects.PartialRepo attribute), 17
 close() (aiogithub.GitHub method), 7
 closed_at (aiogithub.objects.Issue attribute), 23
 closed_at (aiogithub.objects.PullRequest attribute), 21
 collaborators (aiogithub.objects.AuthenticatedUser attribute), 12
 Comment (class in aiogithub.objects), 24
 comments (aiogithub.objects.Issue attribute), 23
 commit (aiogithub.objects.Branch attribute), 18
 commit (aiogithub.objects.Commit attribute), 18
 Commit (class in aiogithub.objects), 18
 commit_id (aiogithub.objects.IssueEvent attribute), 24

commit_id (aiogithub.objects.ReviewComment attribute), 22
 committer (aiogithub.objects.Commit attribute), 18
 company (aiogithub.objects.Organization attribute), 15
 company (aiogithub.objects.User attribute), 12
 created_at (aiogithub.objects.Comment attribute), 24
 created_at (aiogithub.objects.Issue attribute), 23
 created_at (aiogithub.objects.IssueEvent attribute), 24
 created_at (aiogithub.objects.Organization attribute), 15
 created_at (aiogithub.objects.PullRequest attribute), 21
 created_at (aiogithub.objects.Repo attribute), 18
 created_at (aiogithub.objects.ReviewComment attribute), 22
 created_at (aiogithub.objects.User attribute), 12

D

default_branch (aiogithub.objects.Repo attribute), 18
 description (aiogithub.objects.PartialOrganization attribute), 15
 description (aiogithub.objects.PartialRepo attribute), 17
 diff_hunk (aiogithub.objects.ReviewComment attribute), 22
 diff_url (aiogithub.objects.PullRequest attribute), 21
 disk_usage (aiogithub.objects.AuthenticatedUser attribute), 12

E

email (aiogithub.objects.Organization attribute), 15
 email (aiogithub.objects.User attribute), 12
 event (aiogithub.objects.IssueEvent attribute), 24

F

fetch_data() (aiogithub.objects.BaseResponseObject method), 9
 files (aiogithub.objects.Commit attribute), 18
 followers (aiogithub.objects.Organization attribute), 15
 followers (aiogithub.objects.User attribute), 12
 following (aiogithub.objects.Organization attribute), 16
 following (aiogithub.objects.User attribute), 12

fork (aiogithub.objects.PartialRepo attribute), 17
 forks_count (aiogithub.objects.Repo attribute), 18
 full_name (aiogithub.objects.PartialRepo attribute), 17

G

get_assignees() (aiogithub.objects.PartialRepo method), 17
 get_blobs() (aiogithub.objects.PartialRepo method), 17
 get_branch() (aiogithub.GitHub method), 7
 get_branch() (aiogithub.objects.PartialRepo method), 17
 get_branches() (aiogithub.objects.PartialRepo method), 17
 get_collaborators() (aiogithub.objects.PartialRepo method), 17
 get_comments() (aiogithub.objects.Commit method), 19
 get_comments() (aiogithub.objects.Issue method), 23
 get_comments() (aiogithub.objects.PartialRepo method), 17
 get_comments() (aiogithub.objects.PullRequest method), 21
 get_commits() (aiogithub.objects.PartialRepo method), 17
 get_commits() (aiogithub.objects.PullRequest method), 21
 get_contributors() (aiogithub.objects.PartialRepo method), 17
 get_current_user() (aiogithub.GitHub method), 7
 get_events() (aiogithub.objects.Issue method), 23
 get_events() (aiogithub.objects.PartialOrganization method), 15
 get_events() (aiogithub.objects.PartialRepo method), 17
 get_events() (aiogithub.objects.PartialUser method), 11
 get_followers() (aiogithub.objects.PartialOrganization method), 15
 get_followers() (aiogithub.objects.PartialUser method), 11
 get_following() (aiogithub.objects.PartialOrganization method), 15
 get_following() (aiogithub.objects.PartialUser method), 11
 get_forks() (aiogithub.objects.PartialRepo method), 17
 get_gists() (aiogithub.objects.PartialOrganization method), 15
 get_hooks() (aiogithub.objects.PartialOrganization method), 15
 get_issue() (aiogithub.GitHub method), 7
 get_issue() (aiogithub.objects.PullRequest method), 21
 get_issues() (aiogithub.objects.PartialOrganization method), 15
 get_issues() (aiogithub.objects.PartialRepo method), 17
 get_labels() (aiogithub.objects.Issue method), 23
 get_members() (aiogithub.objects.PartialOrganization method), 15
 get_organization() (aiogithub.GitHub method), 7

get_public_members() (aiogithub.objects.PartialOrganization method), 15
 get_pull_request() (aiogithub.GitHub method), 8
 get_pull_request() (aiogithub.objects.ReviewComment method), 22
 get_pull_requests() (aiogithub.objects.PartialRepo method), 17
 get_rate_limit() (aiogithub.GitHub method), 8
 get_received_events() (aiogithub.objects.PartialUser method), 11
 get_repo() (aiogithub.GitHub method), 8
 get_repo() (aiogithub.objects.Issue method), 23
 get_repos() (aiogithub.GitHub method), 8
 get_repos() (aiogithub.objects.PartialOrganization method), 15
 get_repos() (aiogithub.objects.PartialUser method), 11
 get_requested_reviewers() (aiogithub.objects.PullRequest method), 21
 get_review_comment() (aiogithub.objects.PullRequest method), 21
 get_review_comments() (aiogithub.objects.PullRequest method), 21
 get_reviews() (aiogithub.objects.PullRequest method), 21
 get_stargazers() (aiogithub.objects.PartialRepo method), 17
 get_starred() (aiogithub.objects.PartialUser method), 11
 get_subscriptions() (aiogithub.objects.PartialUser method), 11
 get_user() (aiogithub.GitHub method), 8
 get_users() (aiogithub.GitHub method), 8
 git_url (aiogithub.objects.PartialRepo attribute), 17
 GitHub (class in aiogithub), 7
 gravatar_id (aiogithub.objects.PartialUser attribute), 11

H

has_downloads (aiogithub.objects.Repo attribute), 18
 has_issues (aiogithub.objects.Repo attribute), 18
 has_pages (aiogithub.objects.Repo attribute), 18
 has_wiki (aiogithub.objects.Repo attribute), 18
 head (aiogithub.objects.PullRequest attribute), 21
 hireable (aiogithub.objects.User attribute), 12
 homepage (aiogithub.objects.PartialRepo attribute), 17
 html_url (aiogithub.objects.Comment attribute), 24
 html_url (aiogithub.objects.Commit attribute), 19
 html_url (aiogithub.objects.Issue attribute), 23
 html_url (aiogithub.objects.Organization attribute), 16
 html_url (aiogithub.objects.PartialRepo attribute), 17
 html_url (aiogithub.objects.PartialUser attribute), 11
 html_url (aiogithub.objects.PullRequest attribute), 21
 html_url (aiogithub.objects.ReviewComment attribute), 22

I

id (aiogithub.objects.Comment attribute), 24
 id (aiogithub.objects.Issue attribute), 23
 id (aiogithub.objects.IssueEvent attribute), 24
 id (aiogithub.objects.PartialOrganization attribute), 15
 id (aiogithub.objects.PartialRepo attribute), 18
 id (aiogithub.objects.PartialUser attribute), 11
 id (aiogithub.objects.PullRequest attribute), 21
 id (aiogithub.objects.ReviewComment attribute), 22
 issue (aiogithub.objects.IssueEvent attribute), 24
 Issue (class in aiogithub.objects), 23
 IssueEvent (class in aiogithub.objects), 24

L

labels (aiogithub.objects.Issue attribute), 23
 language (aiogithub.objects.Repo attribute), 18
 last_rate_limit (aiogithub.GitHub attribute), 8
 limit() (aiogithub.objects.PaginatedListProxy method), 9
 limits (aiogithub.objects.BaseResponseObject attribute), 9
 links (aiogithub.objects.Branch attribute), 18
 location (aiogithub.objects.Organization attribute), 16
 location (aiogithub.objects.User attribute), 12
 locked (aiogithub.objects.Issue attribute), 23
 locked (aiogithub.objects.PullRequest attribute), 21
 login (aiogithub.objects.PartialOrganization attribute), 15
 login (aiogithub.objects.PartialUser attribute), 11

M

merged_at (aiogithub.objects.PullRequest attribute), 21
 milestone (aiogithub.objects.Issue attribute), 23
 milestone (aiogithub.objects.PullRequest attribute), 21
 mirror_url (aiogithub.objects.PartialRepo attribute), 18

N

name (aiogithub.objects.Branch attribute), 18
 name (aiogithub.objects.Organization attribute), 16
 name (aiogithub.objects.PartialRepo attribute), 18
 number (aiogithub.objects.Issue attribute), 23
 number (aiogithub.objects.PullRequest attribute), 22

O

open_issues_count (aiogithub.objects.Repo attribute), 18
 Organization (class in aiogithub.objects), 15
 original_commit_id (aiogithub.objects.ReviewComment attribute), 22
 original_position (aiogithub.objects.ReviewComment attribute), 22
 owned_private_repos (aiogithub.objects.AuthenticatedUser attribute), 12
 owner (aiogithub.objects.PartialRepo attribute), 18

P

PaginatedListProxy (class in aiogithub.objects), 9
 parents (aiogithub.objects.Commit attribute), 19
 PartialOrganization (class in aiogithub.objects), 15
 PartialRepo (class in aiogithub.objects), 17
 PartialUser (class in aiogithub.objects), 11
 patch_url (aiogithub.objects.PullRequest attribute), 22
 path (aiogithub.objects.ReviewComment attribute), 22
 permissions (aiogithub.objects.Repo attribute), 18
 plan (aiogithub.objects.AuthenticatedUser attribute), 12
 position (aiogithub.objects.ReviewComment attribute), 22
 private (aiogithub.objects.PartialRepo attribute), 18
 private_gists (aiogithub.objects.AuthenticatedUser attribute), 12
 public_gists (aiogithub.objects.Organization attribute), 16
 public_gists (aiogithub.objects.User attribute), 12
 public_repos (aiogithub.objects.Organization attribute), 16
 public_repos (aiogithub.objects.User attribute), 12
 PullRequest (class in aiogithub.objects), 21
 pushed_at (aiogithub.objects.Repo attribute), 18

R

Repo (class in aiogithub.objects), 18
 requested_reviewers (aiogithub.objects.PullRequest attribute), 22
 ReviewComment (class in aiogithub.objects), 22

S

sha (aiogithub.objects.Commit attribute), 19
 site_admin (aiogithub.objects.PartialUser attribute), 11
 size (aiogithub.objects.Repo attribute), 18
 ssh_url (aiogithub.objects.PartialRepo attribute), 18
 stargazers_count (aiogithub.objects.Repo attribute), 18
 state (aiogithub.objects.Issue attribute), 23
 state (aiogithub.objects.PullRequest attribute), 22
 stats (aiogithub.objects.Commit attribute), 19
 svn_url (aiogithub.objects.PartialRepo attribute), 18

T

title (aiogithub.objects.Issue attribute), 23
 title (aiogithub.objects.PullRequest attribute), 22
 total_private_repos (aiogithub.objects.AuthenticatedUser attribute), 13
 type (aiogithub.objects.Organization attribute), 16
 type (aiogithub.objects.PartialUser attribute), 11

U

updated_at (aiogithub.objects.Comment attribute), 24
 updated_at (aiogithub.objects.Issue attribute), 23
 updated_at (aiogithub.objects.Organization attribute), 16
 updated_at (aiogithub.objects.PullRequest attribute), 22

updated_at (aiogithub.objects.Repo attribute), 18
updated_at (aiogithub.objects.ReviewComment attribute), 22
updated_at (aiogithub.objects.User attribute), 12
user (aiogithub.objects.Comment attribute), 24
user (aiogithub.objects.Issue attribute), 23
user (aiogithub.objects.PullRequest attribute), 22
user (aiogithub.objects.ReviewComment attribute), 22
User (class in aiogithub.objects), 11

W

watchers_count (aiogithub.objects.Repo attribute), 18