The REST API is now versioned. For more information, see "About API versioning."

Organizations

Use the REST API to interact with organizations.

List organizations

✓ Works with GitHub Apps

Lists all organizations, in the order that they were created on GitHub Enterprise Cloud.

Note: Pagination is powered exclusively by the `since` parameter. Use the Link header to get the URL for the next page of organizations.

Parameters for "List organizations"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Query parameters

`since` integer

An organization ID. Only return organizations with an ID greater than this ID.

`per_page` integer

The number of results per page (max 100).

Default: `30`

HTTP response status codes for "List organizations"

Status code	Description
`200`	OK
`304`	Not modified

Code samples for "List organizations"

GET `/organizations`

cURL JavaScript GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/organizations
```

Response

Example response   Response schema

```
Status: 200
```

```
[ { "login": "github", "id": 1, "node_id": "MDEyOk9yZ2FuaXphdGlvbjE=", "url": "https://api.github.com/orgs/github", "repos_url":
"https://api.github.com/orgs/github/repos", "events_url": "https://api.github.com/orgs/github/events", "hooks_url":
"https://api.github.com/orgs/github/hooks", "issues_url": "https://api.github.com/orgs/github/issues", "members_url":
"https://api.github.com/orgs/github/members{/member}", "public_members_url":
"https://api.github.com/orgs/github/public_members{/member}", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"description": "A great organization" } ]
```

Get an organization

✓ Works with GitHub Apps

To see many of the organization response values, you need to be an authenticated organization owner with the `admin:org` scope. When the value of `two_factor_requirement_enabled` is `true`, the organization requires all members, billing managers, and outside collaborators to enable two-factor authentication.

GitHub Apps with the `Organization plan` permission can use this endpoint to retrieve information about an organization's GitHub Enterprise Cloud plan. See "Authenticating with GitHub Apps" for details. For an example response, see 'Response with GitHub Enterprise Cloud plan information' below."

Parameters for "Get an organization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

HTTP response status codes for "Get an organization"

Status code	Description
`200`	OK
`404`	Resource not found

Code samples for "Get an organization"

GET /orgs/{org}

cURL JavaScript GitHub CLI

```
 curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/orgs/ORG
```

Response

Example response Response schema

Status: 200

```
 { "login": "github", "id": 1, "node_id": "MDEyOk9yZ2FuaXphdGlvbjE=", "url": "https://api.github.com/orgs/github", "repos_url":
"https://api.github.com/orgs/github/repos", "events_url": "https://api.github.com/orgs/github/events", "hooks_url":
"https://api.github.com/orgs/github/hooks", "issues_url": "https://api.github.com/orgs/github/issues", "members_url":
"https://api.github.com/orgs/github/members{/member}", "public_members_url":
"https://api.github.com/orgs/github/public_members{/member}", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"description": "A great organization", "name": "github", "company": "GitHub", "blog": "https://github.com/blog", "location": "San
```

Update an organization

✓ Works with GitHub Apps

Parameter Deprecation Notice: GitHub Enterprise Cloud will replace and discontinue `members_allowed_repository_creation_type` in favor of more granular permissions. The new input parameters are `members_can_create_public_repositories`, `members_can_create_private_repositories` for all organizations and `members_can_create_internal_repositories` for organizations associated with an enterprise account using GitHub Enterprise Cloud or GitHub Enterprise Server 2.20+. For more information, see the blog post.

Enables an authenticated organization owner with the `admin:org` scope or the `repo` scope to update the organization's profile and member privileges.

Parameters for "Update an organization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

Body parameters

`billing_email` string

Billing email address. This address is not publicized.

`company` string

The company name.

`email` string

The publicly visible email address.

`twitter_username` string

The Twitter username of the company.

`location` string

The location.

`name` string

The shorthand name of the company.

`description` string

The description of the company.

`has_organization_projects` boolean

Whether an organization can use organization projects.

`has_repository_projects` boolean

Whether repositories that belong to the organization can use repository projects.

`default_repository_permission` string

Default permission level members have for organization repositories.

Default: `read`
Can be one of: `read` , `write` , `admin` , `none`

`members_can_create_repositories` boolean

Whether of non-admin organization members can create repositories. **Note:** A parameter can override this parameter. See `members_allowed_repository_creation_type` in this table for details.

Default: `true`

`members_can_create_internal_repositories` boolean

Whether organization members can create internal repositories, which are visible to all enterprise members. You can only allow members to create internal repositories if your organization is associated with an enterprise account using GitHub Enterprise Cloud or GitHub Enterprise Server 2.20+. For more information, see "Restricting repository creation in your organization" in the GitHub Help documentation.

`members_can_create_private_repositories` boolean

Whether organization members can create private repositories, which are visible to organization members with permission. For more information, see "Restricting repository creation in your organization" in the GitHub Help documentation.

`members_can_create_public_repositories` boolean

Whether organization members can create public repositories, which are visible to anyone. For more information, see "Restricting repository creation in your organization" in the GitHub Help documentation.

`members_allowed_repository_creation_type` string

Specifies which types of repositories non-admin organization members can create. `private` is only available to repositories that are part of an organization on GitHub Enterprise Cloud. **Note:** This parameter is deprecated and will be removed in the future. Its return value ignores internal repositories. Using this parameter overrides values set in `members_can_create_repositories` . See the parameter deprecation notice in the operation description for details.

Can be one of: `all` , `private` , `none`

`members_can_create_pages` boolean

Whether organization members can create GitHub Pages sites. Existing published sites will not be impacted.

Default: `true`

`members_can_create_public_pages` boolean

Whether organization members can create public GitHub Pages sites. Existing published sites will not be impacted.

Default: `true`

`members_can_create_private_pages` boolean

Whether organization members can create private GitHub Pages sites. Existing published sites will not be impacted.

Default: `true`

`members_can_fork_private_repositories` boolean

Whether organization members can fork private organization repositories.

Default: `false`

`web_commit_signoff_required` boolean

Whether contributors to organization repositories are required to sign off on commits they make through GitHub's web interface.

Default: `false`

`blog` string

`advanced_security_enabled_for_new_repositories` boolean

Whether GitHub Advanced Security is automatically enabled for new repositories.
To use this parameter, you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "Managing security managers in your organization."
You can check which security and analysis features are currently enabled by using a `GET /orgs/{org}` request.

`dependabot_alerts_enabled_for_new_repositories` boolean

Whether Dependabot alerts is automatically enabled for new repositories.
To use this parameter, you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "Managing security managers in your organization."
You can check which security and analysis features are currently enabled by using a `GET /orgs/{org}` request.

`dependabot_security_updates_enabled_for_new_repositories` boolean

Whether Dependabot security updates is automatically enabled for new repositories.
To use this parameter, you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "Managing security managers in your organization."
You can check which security and analysis features are currently enabled by using a `GET /orgs/{org}` request.

`dependency_graph_enabled_for_new_repositories` boolean

Whether dependency graph is automatically enabled for new repositories.
To use this parameter, you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "Managing security managers in your organization."
You can check which security and analysis features are currently enabled by using a `GET /orgs/{org}` request.

`secret_scanning_enabled_for_new_repositories` boolean

Whether secret scanning is automatically enabled for new repositories.
To use this parameter, you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "Managing security managers in your organization."
You can check which security and analysis features are currently enabled by using a `GET /orgs/{org}` request.

`secret_scanning_push_protection_enabled_for_new_repositories` boolean

Whether secret scanning push protection is automatically enabled for new repositories.
To use this parameter, you must have admin permissions for the repository or be an owner or security manager for the organization that owns the repository. For more information, see "Managing security managers in your organization."
You can check which security and analysis features are currently enabled by using a `GET /orgs/{org}` request.

`secret_scanning_push_protection_custom_link_enabled` boolean

Whether a custom link is shown to contributors who are blocked from pushing a secret by push protection.

`secret_scanning_push_protection_custom_link` string

If `secret_scanning_push_protection_custom_link_enabled` is true, the URL that will be displayed to contributors who are blocked from pushing a secret.

HTTP response status codes for "Update an organization"

Status code	Description
`200`	OK
`409`	Conflict
`422`	Validation failed

Code samples for "Update an organization"

PATCH `/orgs/{org}`

cURL JavaScript GitHub CLI

```
curl -L \ -X PATCH \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-
11-28" \ https://api.github.com/orgs/ORG \ -d
'{"billing_email":"mona@github.com","company":"GitHub","email":"mona@github.com","twitter_username":"github","location":"San
Francisco","name":"github","description":"GitHub, the
company.","default_repository_permission":"read","members_can_create_repositories":true,"members_allowed_repository_creation_type":"all"}'
```

Response

Example response Response schema

`Status: 200`

```
{ "login": "github", "id": 1, "node_id": "MDEyOk9yZ2FuaXphdGlvbjE=", "url": "https://api.github.com/orgs/github", "repos_url":
"https://api.github.com/orgs/github/repos", "events_url": "https://api.github.com/orgs/github/events", "hooks_url":
"https://api.github.com/orgs/github/hooks", "issues_url": "https://api.github.com/orgs/github/issues", "members_url":
"https://api.github.com/orgs/github/members{/member}", "public_members_url":
"https://api.github.com/orgs/github/public_members{/member}", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"description": "A great organization", "name": "github", "company": "GitHub", "blog": "https://github.com/blog", "location": "San
```

Delete an organization

Works with GitHub Apps

Deletes an organization and all its repositories.

The organization login will be unavailable for 90 days after deletion.

Please review the Terms of Service regarding account deletion before using this endpoint:

https://docs.github.com/enterprise-cloud@latest//site-policy/github-terms/github-terms-of-service

Parameters for "Delete an organization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

HTTP response status codes for "Delete an organization"

Status code	Description
`202`	Accepted
`403`	Forbidden
`404`	Resource not found

Code samples for "Delete an organization"

DELETE `/orgs/{org}`

cURL JavaScript GitHub CLI

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/orgs/ORG
```

Accepted

Example response Response schema

```
Status: 202
```

Get the audit log for an organization

✓ Works with [GitHub Apps](#)

Gets the audit log for an organization. For more information, see "[Reviewing the audit log for your organization](#)."

To use this endpoint, you must be an organization owner, and you must use an access token with the `read:audit_log` scope. GitHub Apps must have the `organization_administration` read permission to use this endpoint.

By default, the response includes up to 30 events from the past three months. Use the `phrase` parameter to filter results and retrieve older events. For example, use the `phrase` parameter with the `created` qualifier to filter events based on when the events occurred. For more information, see "[Reviewing the audit log for your organization](#)."

Use pagination to retrieve fewer or more than 30 events. For more information, see "[Resources in the REST API](#)."

This endpoint has a rate limit of 1,750 queries per hour per user and IP address. If your integration receives a rate limit error (typically a 403 or 429 response), it should wait before making another request to the GitHub API. For more information, see "[Resources in the REST API](#)" and "[Best practices for integrators](#)."

Parameters for "Get the audit log for an organization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

Query parameters

`phrase` string

A search phrase. For more information, see [Searching the audit log](#).

`include` string

The event types to include:

- `web` - returns web (non-Git) events.
- `git` - returns Git events.
- `all` - returns both web and Git events.

The default is `web`.

Can be one of: `web`, `git`, `all`

`after` string

A cursor, as given in the [Link header](#). If specified, the query only searches for events after this cursor.

`before` string

A cursor, as given in the [Link header](#). If specified, the query only searches for events before this cursor.

`order` string

The order of audit log events. To list newest events first, specify `desc`. To list oldest events first, specify `asc`.
The default is `desc`.

Can be one of: `desc`, `asc`

`per_page` integer

The number of results per page (max 100).

Default: `30`

HTTP response status codes for "Get the audit log for an organization"

Status code	Description
`200`	OK

Code samples for "Get the audit log for an organization"

GET `/orgs/{org}/audit-log`

cURL JavaScript GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/orgs/ORG/audit-log
```

Response

Example response Response schema

```
Status: 200
```

```
[ { "@timestamp": 1606929874512, "action": "team.add_member", "actor": "octocat", "created_at": 1606929874512, "_document_id":
"xJJFlFOhQ6b-5vaAFy9Rjw", "org": "octo-corp", "team": "octo-corp/example-team", "user": "monalisa" }, { "@timestamp": 1606507117008,
"action": "org.create", "actor": "octocat", "created_at": 1606507117008, "_document_id": "Vqvg6kZ4MYqwWRKFDzlMoQ", "org": "octocat-test-
org" }, { "@timestamp": 1605719148837, "action": "repo.destroy", "actor": "monalisa", "created_at": 1605719148837, "_document_id":
"LwW2vpJZCDS-WUmo9Z-ifw", "org": "mona-org", "repo": "mona-org/mona-test-repo", "visibility": "private" } ]
```

List SAML SSO authorizations for an organization

✓ Works with GitHub Apps

Listing and deleting credential authorizations is available to organizations with GitHub Enterprise Cloud. For more information, see GitHub's products.

An authenticated organization owner with the `read:org` scope can list all credential authorizations for an organization that uses SAML single sign-on (SSO). The credentials are either personal access tokens or SSH keys that organization members have authorized for the organization. For more information, see About authentication with SAML single sign-on.

Parameters for "List SAML SSO authorizations for an organization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

Query parameters

`per_page` integer

The number of results per page (max 100).

Default: `30`

`page` integer

Page token

`login` string

Limits the list of credentials authorizations for an organization to a specific login

HTTP response status codes for "List SAML SSO authorizations for an organization"

Status code	Description

200	OK

Code samples for "List SAML SSO authorizations for an organization"

GET /orgs/{org}/credential-authorizations

cURL JavaScript GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/orgs/ORG/credential-authorizations
```

Response

Example response Response schema

Status: 200

```
[ { "login": "octocat", "credential_id": 161195, "credential_type": "personal access token", "token_last_eight": "71c3fc11",
"credential_authorized_at": "2011-01-26T19:06:43Z", "credential_accessed_at": "2011-01-26T19:06:43Z", "authorized_credential_expires_at":
"2011-02-25T19:06:43Z", "scopes": [ "user", "repo" ] }, { "login": "hubot", "credential_id": 161196, "credential_type": "personal access
token", "token_last_eight": "Ae178B4a", "credential_authorized_at": "2019-03-29T19:06:43Z", "credential_accessed_at": "2011-01-
26T19:06:43Z", "authorized_credential_expires_at": "2019-04-28T19:06:43Z", "scopes": [ "repo" ] } ]
```

Remove a SAML SSO authorization for an organization

Works with GitHub Apps

Listing and deleting credential authorizations is available to organizations with GitHub Enterprise Cloud. For more information, see GitHub's products.

An authenticated organization owner with the `admin:org` scope can remove a credential authorization for an organization that uses SAML SSO. Once you remove someone's credential authorization, they will need to create a new personal access token or SSH key and authorize it for the organization they want to access.

Parameters for "Remove a SAML SSO authorization for an organization"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

org string Required

The organization name. The name is not case sensitive.

credential_id integer Required

HTTP response status codes for "Remove a SAML SSO authorization for an organization"

Status code	Description

`204`	No Content
`404`	Resource not found

Code samples for "Remove a SAML SSO authorization for an organization"

DELETE `/orgs/{org}/credential-authorizations/{credential_id}`

cURL JavaScript GitHub CLI

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version:
2022-11-28" \ https://api.github.com/orgs/ORG/credential-authorizations/CREDENTIAL_ID
```

Response

```
Status: 204
```

List app installations for an organization

✓ Works with GitHub Apps

Lists all GitHub Apps in an organization. The installation count includes all GitHub Apps installed on repositories in the organization. You must be an organization owner with `admin:read` scope to use this endpoint.

Parameters for "List app installations for an organization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

Query parameters

`per_page` integer

The number of results per page (max 100).

Default: `30`

`page` integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List app installations for an organization"

Status code	Description
`200`	OK

Code samples for "List app installations for an organization"

GET `/orgs/{org}/installations`

cURL JavaScript GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/orgs/ORG/installations
```

Response

Example response Response schema

Status: 200

```
{ "total_count": 1, "installations": [ { "id": 25381, "account": { "login": "octo-org", "id": 6811672, "node_id":
"MDEyOk9yZ2FuaXphdGlvbjY4MTE2NzI=", "avatar_url": "https://avatars3.githubusercontent.com/u/6811672?v=4", "gravatar_id": "", "url":
"https://api.github.com/users/octo-org", "html_url": "https://github.com/octo-org", "followers_url": "https://api.github.com/users/octo-
org/followers", "following_url": "https://api.github.com/users/octo-org/following{/other_user}", "gists_url":
"https://api.github.com/users/octo-org/gists{/gist_id}", "starred_url": "https://api.github.com/users/octo-org/starred{/owner}{/repo}",
"subscriptions_url": "https://api.github.com/users/octo-org/subscriptions", "organizations_url": "https://api.github.com/users/octo-
```

Enable or disable a security feature for an organization

✓ Works with GitHub Apps

Enables or disables the specified security feature for all eligible repositories in an organization.

To use this endpoint, you must be an organization owner or be member of a team with the security manager role. A token with the 'write:org' scope is also required.

GitHub Apps must have the `organization_administration:write` permission to use this endpoint.

For more information, see "Managing security managers in your organization."

Parameters for "Enable or disable a security feature for an organization"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

`security_product` string Required

The security feature to enable or disable.

Can be one of: `dependency_graph`, `dependabot_alerts`, `dependabot_security_updates`, `advanced_security`, `code_scanning_default_setup`, `secret_scanning`, `secret_scanning_push_protection`

`enablement` string Required

The action to take.
`enable_all` means to enable the specified security feature for all repositories in the organization. `disable_all` means to disable the specified security feature for all repositories in the organization.

Can be one of: `enable_all`, `disable_all`

Body parameters

`query_suite` string

CodeQL query suite to be used. If you specify the `query_suite` parameter, the default setup will be configured with this query suite only on all repositories that didn't have default setup already configured. It will not change the query suite on repositories that already have default setup configured. If you don't specify any `query_suite` in your request, the preferred query suite of the organization will be applied.

Can be one of: `default`, `extended`

HTTP response status codes for "Enable or disable a security feature for an organization"

Status code	Description
`204`	Action started
`422`	The action could not be taken due to an in progress enablement, or a policy is preventing enablement

Code samples for "Enable or disable a security feature for an organization"

POST `/orgs/{org}/{security_product}/{enablement}`

cURL JavaScript GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ https://api.github.com/orgs/ORG/SECURITY_PRODUCT/ENABLEMENT \ -d '{"query_suite":"default"}'
```

Action started

```
Status: 204
```

List organizations for the authenticated user

List organizations for the authenticated user.

OAuth scope requirements

This only lists organizations that your authorization allows you to operate on in some way (e.g., you can list teams with `read:org` scope, you can publicize your organization membership with `user` scope, etc.). Therefore, this API requires at least `user` or `read:org` scope. OAuth requests with insufficient scope receive a `403 Forbidden` response.

Parameters for "List organizations for the authenticated user"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Query parameters

per_page integer

The number of results per page (max 100).

Default: `30`

page integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List organizations for the authenticated user"

Status code	Description
`200`	OK
`304`	Not modified
`401`	Requires authentication
`403`	Forbidden

Code samples for "List organizations for the authenticated user"

GET /user/orgs

cURL JavaScript GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/user/orgs
```

Response

Example response Response schema

Status: 200

```
[ { "login": "github", "id": 1, "node_id": "MDEyOk9yZ2FuaXphdGlvbjE=", "url": "https://api.github.com/orgs/github", "repos_url":
"https://api.github.com/orgs/github/repos", "events_url": "https://api.github.com/orgs/github/events", "hooks_url":
"https://api.github.com/orgs/github/hooks", "issues_url": "https://api.github.com/orgs/github/issues", "members_url":
"https://api.github.com/orgs/github/members{/member}", "public_members_url":
"https://api.github.com/orgs/github/public_members{/member}", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"description": "A great organization" } ]
```

List organizations for a user

✔ Works with GitHub Apps

List public organization memberships for the specified user.

This method only lists *public* memberships, regardless of authentication. If you need to fetch all of the organization memberships (public and private) for the authenticated user, use the List organizations for the authenticated user API instead.

Parameters for "List organizations for a user"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

username string Required

The handle for the GitHub user account.

Query parameters

per_page integer

The number of results per page (max 100).

Default: `30`

page integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List organizations for a user"

Status code	Description
`200`	OK

Code samples for "List organizations for a user"

GET /users/{username}/orgs

cURL JavaScript GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
https://api.github.com/users/USERNAME/orgs
```

Response

Example response Response schema

```
Status: 200
```

```
[ { "login": "github", "id": 1, "node_id": "MDEyOk9yZ2FuaXphdGlvbjE=", "url": "https://api.github.com/orgs/github", "repos_url":
```

```
"https://api.github.com/orgs/github/repos", "events_url": "https://api.github.com/orgs/github/events", "hooks_url":
"https://api.github.com/orgs/github/hooks", "issues_url": "https://api.github.com/orgs/github/issues", "members_url":
"https://api.github.com/orgs/github/members{/member}", "public_members_url":
"https://api.github.com/orgs/github/public_members{/member}", "avatar_url": "https://github.com/images/error/octocat_happy.gif",
"description": "A great organization" } ]
```