

The REST API is now versioned. For more information, see "[About API versioning](#)."

Git trees

Use the REST API to interact with tree objects in your Git database on GitHub Enterprise Cloud.

About Git trees

A Git tree object creates the hierarchy between files in a Git repository. You can use the Git tree object to create the relationship between directories and the files they contain. These endpoints allow you to read and write [tree objects](#) to your Git database on GitHub Enterprise Cloud.

Create a tree

✔ Works with [GitHub Apps](#)

The tree creation API accepts nested entries. If you specify both a tree and a nested path modifying that tree, this endpoint will overwrite the contents of the tree with the new path contents, and create a new tree structure.

If you use this endpoint to add, delete, or modify the file contents in a tree, you will need to commit the tree and then update a branch to point to the commit. For more information see "[Create a commit](#)" and "[Update a reference](#)."

Returns an error if you try to delete a file that does not exist.

Parameters for "Create a tree"

Headers

accept string

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

Path parameters

owner string **Required**

The account owner of the repository. The name is not case sensitive.

repo string **Required**

The name of the repository without the `.git` extension. The name is not case sensitive.

Body parameters

tree array of objects **Required**

Objects (of `path` , `mode` , `type` , and `sha`) specifying a tree structure.

► Properties of `tree`

base_tree `string`

The SHA1 of an existing Git tree object which will be used as the base for the new tree. If provided, a new Git tree object will be created from entries in the Git tree object pointed to by `base_tree` and entries defined in the `tree` parameter. Entries defined in the `tree` parameter will overwrite items from `base_tree` with the same `path` . If you're creating new changes on a branch, then normally you'd set `base_tree` to the SHA1 of the Git tree object of the current latest commit on the branch you're working on. If not provided, GitHub will create a new Git tree object from only the entries defined in the `tree` parameter. If you create a new commit pointing to such a tree, then all files which were a part of the parent commit's tree and were not defined in the `tree` parameter will be listed as deleted by the new commit.

HTTP response status codes for "Create a tree"

Status code	Description
201	Created
403	Forbidden
404	Resource not found
422	Validation failed, or the endpoint has been spammed.

Code samples for "Create a tree"

POST /repos/{owner}/{repo}/git/trees

cURL

JavaScript

GitHub CLI

Copy icon

```
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/repos/OWNER/REPO/git/trees \ -d '{"base_tree":"9fb037999f264ba9a7fc6274d15fa3ae2ab98312","tree":[{"path":"file.rb","mode":"100644","type":"blob","sha":"44b4fc6d56897b048c772eb4087f854f46256132"}]}'
```

Response

Example response

Response schema

Status: 201

```
{ "sha": "cd8274d15fa3ae2ab983129fb037999f264ba9a7", "url": "https://api.github.com/repos/octocat/Hello-World/trees/cd8274d15fa3ae2ab983129fb037999f264ba9a7", "tree": [ { "path": "file.rb", "mode": "100644", "type": "blob", "size": 132, "sha": "7c258a9869f33c1e1f74fbb32f07c86cb5a75b", "url": "https://api.github.com/repos/octocat/Hello-World/git/blobs/7c258a9869f33c1e1f74fbb32f07c86cb5a75b" } ], "truncated": true }
```

Get a tree 🔗

✔ Works with [GitHub Apps](#)

Returns a single tree using the SHA1 value or ref name for that tree.

If `truncated` is `true` in the response then the number of items in the `tree` array exceeded our maximum limit. If you need to fetch more items, use the non-recursive method of fetching trees, and fetch one sub-tree at a time.

Note: The limit for the `tree` array is 100,000 entries with a maximum size of 7 MB when using the `recursive` parameter.

Parameters for "Get a tree"

Headers

`accept` string

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

Path parameters

`owner` string **Required**

The account owner of the repository. The name is not case sensitive.

`repo` string **Required**

The name of the repository without the `.git` extension. The name is not case sensitive.

`tree_sha` string **Required**

The SHA1 value or ref (branch or tag) name of the tree.

Query parameters

`recursive` string

Setting this parameter to any value returns the objects or subtrees referenced by the tree specified in `:tree_sha`. For example, setting `recursive` to any of the following will enable returning objects or subtrees: `0`, `1`, `"true"`, and `"false"`. Omit this parameter to prevent recursively returning objects or subtrees.

HTTP response status codes for "Get a tree"

Status code	Description
200	OK
404	Resource not found
422	Validation failed, or the endpoint has been spammed.

Code samples for "Get a tree"

Example 1: Status Code 200 ▾

GET

/repos/{owner}/{repo}/git/trees/{tree_sha}

cURL

JavaScript

GitHub CLI

Copy icon

```
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/repos/OWNER/REPO/git/trees/TREE_SHA
```

Default response

Example response

Response schema

Status: 200

```
{ "sha": "9fb037999f264ba9a7fc6274d15fa3ae2ab98312", "url": "https://api.github.com/repos/octocat/Hello-World/trees/9fb037999f264ba9a7fc6274d15fa3ae2ab98312", "tree": [ { "path": "file.rb", "mode": "100644", "type": "blob", "size": 30, "sha": "44b4fc6d56897b048c772eb4087f854f46256132", "url": "https://api.github.com/repos/octocat/Hello-World/git/blobs/44b4fc6d56897b048c772eb4087f854f46256132" }, { "path": "subdir", "mode": "040000", "type": "tree", "sha": "f484d249c660418515fb01c2b9662073663c242e", "url": "https://api.github.com/repos/octocat/Hello-World/git/blobs/f484d249c660418515fb01c2b9662073663c242e" }, { "path": "exec_file", "mode": "100755", "type": "blob", "size": 75,
```

Legal