

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

# Git references

Use the REST API to interact with references in your Git database on GitHub Enterprise Server

## About Git references

A Git reference ( `git ref` ) is a file that contains a Git commit SHA-1 hash. When referring to a Git commit, you can use the Git reference, which is an easy-to-remember name, rather than the hash. The Git reference can be rewritten to point to a new commit. A branch is a Git reference that stores the new Git commit hash. These endpoints allow you to read and write [references](#) to your Git database on GitHub Enterprise Server.

## List matching references

✔ Works with [GitHub Apps](#)

Returns an array of references from your Git database that match the supplied name. The `:ref` in the URL must be formatted as `heads/<branch name>` for branches and `tags/<tag name>` for tags. If the `:ref` doesn't exist in the repository, but existing refs start with `:ref`, they will be returned as an array.

When you use this endpoint without providing a `:ref`, it will return an array of all the references from your Git database, including notes and stashes if they exist on the server. Anything in the namespace is returned, not just `heads` and `tags`.

**Note:** You need to explicitly [request a pull request](#) to trigger a test merge commit, which checks the mergeability of pull requests. For more information, see "[Checking mergeability of pull requests](#)".

If you request matching references for a branch named `feature` but the branch `feature` doesn't exist, the response can still include other matching head refs that start with the word `feature`, such as `featureA` and `featureB`.

### Parameters for "List matching references"

#### 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.

**ref** string **Required**

The commit reference. Can be a commit SHA, branch name ( `heads/BRANCH_NAME` ), or tag name ( `tags/TAG_NAME` ). For more information, see "[Git References](#)" in the Git documentation.

## HTTP response status codes for "List matching references"

Status code	Description
200	OK

## Code samples for "List matching references"

**GET** /repos/{owner}/{repo}/git/matching-refs/{ref}

cURLJavaScriptGitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/git/matching-refs/REF
```

## Response

Example responseResponse schema

Status: 200

```
[ { "ref": "refs/heads/feature-a", "node_id": "MDM6UmVmcmVmcy9oZWFKcy9mZWFOdXJlLWE=", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/refs/heads/feature-a", "object": { "type": "commit", "sha": "aa218f56b14c9653891f9e74264a383fa43fefbd", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/commits/aa218f56b14c9653891f9e74264a383fa43fefbd" } }, { "ref": "refs/heads/feature-b", "node_id": "MDM6UmVmcmVmcy9oZWFKcy9mZWFOdXJlLWE=", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/refs/heads/feature-b", "object": { "type": "commit", "sha": "612077ae6dffb4d2fbd8ce0cccaa58893b07b5ac", "url": "https://HOSTNAME/repos/octocat/Hello-
```

## Get a reference

✔ Works with [GitHub Apps](#)

Returns a single reference from your Git database. The `:ref` in the URL must be formatted as `heads/<branch name>` for branches and `tags/<tag name>` for tags. If the `:ref` doesn't match an existing ref, a `404` is returned.

**Note:** You need to explicitly [request a pull request](#) to trigger a test merge commit, which checks the mergeability of pull requests. For more information, see "[Checking mergeability of pull requests](#)".

## Parameters for "Get a reference"

**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.

**ref** string **Required**

The commit reference. Can be a commit SHA, branch name ( `heads/BRANCH_NAME` ), or tag name ( `tags/TAG_NAME` ). For more information, see "[Git References](#)" in the Git documentation.

HTTP response status codes for "Get a reference"

Status code	Description
200	OK
404	Resource not found

Code samples for "Get a reference"

GET

/repos/{owner}/{repo}/git/ref/{ref}

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" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/git/ref/REF
```

Response

Example response

Response schema

Status: 200

```
{ "ref": "refs/heads/featureA", "node_id": "MDM6UmVmcmVmcy9oZWFKcy9mZW0dXJlQQ==", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/refs/heads/featureA", "object": { "type": "commit", "sha": "aa218f56b14c9653891f9e74264a383fa43fefbd", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/commits/aa218f56b14c9653891f9e74264a383fa43fefbd" } }
```

Create a reference [↗](#)

✔ Works with [GitHub Apps](#)

Create a reference for your repository. You are unable to create new references for empty repositories, even if the

Creates a reference for your repository. You are unable to create new references for empty repositories, even if the commit SHA-1 hash used exists. Empty repositories are repositories without branches.

Parameters for "Create a reference"

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

**ref** string **Required**  
The name of the fully qualified reference (ie: `refs/heads/master` ). If it doesn't start with 'refs' and have at least two slashes, it will be rejected.

**sha** string **Required**  
The SHA1 value for this reference.

HTTP response status codes for "Create a reference"

Status code	Description
201	Created
422	Validation failed, or the endpoint has been spammed.

Code samples for "Create a reference"

POST

`/repos/{owner}/{repo}/git/refs`

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" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/git/refs \ -d '{ "ref": "refs/heads/featureA", "sha": "aa218f56b14c9653891f9e74264a383fa43fefbd" }'
```

Response

Example response

Response schema

Status: 201

```
{ "ref": "refs/heads/featureA", "node_id": "MDM6UmVmcmVmcy9oZWFKcy9mZWFOdXJlQQ==", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/refs/heads/featureA", "object": { "type": "commit", "sha": "aa218f56b14c9653891f9e74264a383fa43fefbd", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/commits/aa218f56b14c9653891f9e74264a383fa43fefbd" } }
```

# Update a reference

✔ Works with [GitHub Apps](#)

## Parameters for "Update a reference"

### 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.

**ref** string **Required**  
The name of the reference to update (for example, `heads/featureA` ). Can be a branch name ( `heads/BRANCH_NAME` ) or tag name ( `tags/TAG_NAME` ). For more information, see "[Git References](#)" in the Git documentation.

### Body parameters

**sha** string **Required**  
The SHA1 value to set this reference to

**force** boolean  
Indicates whether to force the update or to make sure the update is a fast-forward update. Leaving this out or setting it to `false` will make sure you're not overwriting work.  
Default: `false`

## HTTP response status codes for "Update a reference"

Status code	Description
200	OK
422	Validation failed, or the endpoint has been spammed.

## Code samples for "Update a reference"

PATCH /repos/{owner}/{repo}/git/refs/{ref}

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" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/git/refs/REF \ -d '{ "sha": "aa218f56b14c9653891f9e74264a383fa43fefbd", "force": true }'
```

Response

Example response

Response schema

Status: 200

{ "ref": "refs/heads/featureA", "node\_id": "MDM6UmVmcmVmcy9oZWFKcy9mZWdXJlQQ==", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/refs/heads/featureA", "object": { "type": "commit", "sha": "aa218f56b14c9653891f9e74264a383fa43fefbd", "url": "https://HOSTNAME/repos/octocat/Hello-World/git/commits/aa218f56b14c9653891f9e74264a383fa43fefbd" } }

## Delete a reference

✔ Works with [GitHub Apps](#)

### Parameters for "Delete a reference"

#### 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.

**ref** string **Required**

The commit reference. Can be a commit SHA, branch name ( `heads/BRANCH_NAME` ), or tag name ( `tags/TAG_NAME` ). For more information, see "[Git References](#)" in the Git documentation.

### HTTP response status codes for "Delete a reference"

Status code	Description
204	Successful deletion.

Code samples for "Delete a reference"

DELETE

/repos/{owner}/{repo}/git/refs/{ref}

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" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/git/refs/REF
```

Response

Status: 204

Legal