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

# **Commits**

GitHub Docs

Use the REST API to interact with commits.

# **List commits** *∂*

Works with <u>GitHub Apps</u>

# Signature verification object

The response will include a verification object that describes the result of verifying the commit's signature. The following fields are included in the verification object:

Name	Туре	Description
verified	boolean	Indicates whether GitHub considers the signature in this commit to be verified.
reason	string	The reason for verified value. Possible values and their meanings are enumerated in table below.
signature	string	The signature that was extracted from the commit.
payload	string	The value that was signed.

These are the possible values for reason in the verification object:

Value	Description
expired_key	The key that made the signature is expired.
not_signing_key	The "signing" flag is not among the usage flags in the GPG key that made the signature.
<pre>gpgverify_error</pre>	There was an error communicating with the signature verification service.
gpgverify_unavailable	The signature verification service is currently unavailable.
unsigned	The object does not include a signature.
unknown_signature_type	A non-PGP signature was found in the commit.
no_user	No user was associated with the committer email address in the commit.
unverified_email	The committee email address in the commit was associated with a user, but the email address is not verified on their account.

bad_email	The committer email address in the commit is not included in the identities of the PGP key that made the signature.
unknown_key	The key that made the signature has not been registered with any user's account.
malformed_signature	There was an error parsing the signature.
invalid	The signature could not be cryptographically verified using the key whose key-id was found in the signature.
valid	None of the above errors applied, so the signature is considered to be verified.

#### Parameters for "List commits"

### **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 <code>.git</code> extension. The name is not case sensitive.

# **Query parameters**

**sha** string

SHA or branch to start listing commits from. Default: the repository's default branch (usually  $_{ t main}$ ).

path string

Only commits containing this file path will be returned.

author string

GitHub username or email address to use to filter by commit author.

committer string

GitHub username or email address to use to filter by commit committer.

since string

Only show results that were last updated after the given time. This is a timestamp in ISO 8601 format: YYYY-MM-DDTHH:MM:SSZ.

until string

Only commits before this date will be returned. This is a timestamp in ISO 8601 format: YYYY-MM-DDTHH:MM:SSZ.

per\_page integer

The number of results per page (max 100).

. . -

page integer

Default: 30

Page number of the results to fetch.

Default: 1

### HTTP response status codes for "List commits"

Status code	Description
200	OK
400	Bad Request
404	Resource not found
409	Conflict
500	Internal Error

#### Code samples for "List commits"



### Response

Example response Response schema

Status: 200

[{ "url": "https://api.github.com/repos/octocat/Hello-World/commits/6dcb09b5b57875f334f61aebed695e2e4193db5e", "sha": "6dcb09b5b57875f334f61aebed695e2e4193db5e", "node\_id": "MDY6029tbWl0NmRjYjA5YjViNTc4NzVmMzM0ZjYxYWViZWQ20TVlMmU0MTkzZGI1ZQ==", "html\_url": "https://github.com/octocat/Hello-World/commit/6dcb09b5b57875f334f61aebed695e2e4193db5e", "comments\_url": "https://api.github.com/repos/octocat/Hello-World/commits/6dcb09b5b57875f334f61aebed695e2e4193db5e/comments", "commit": { "url": "https://api.github.com/repos/octocat/Hello-World/git/commits/6dcb09b5b57875f334f61aebed695e2e4193db5e", "author": { "name": "Monalisa Octocat", "email": "support@github.com", "date": "2011-04-14T16:00:49Z" }, "committer": { "name": "Monalisa Octocat", "

# List branches for HEAD commit @

### Works with <u>GitHub Apps</u>

Protected branches are available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server. For more information, see <u>GitHub's products</u> in the GitHub Help documentation.

Returns all branches where the given commit SHA is the HEAD, or latest commit for the branch.

### Parameters for "List branches for HEAD commit"

#### **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.

commit sha string Required

The SHA of the commit.

# HTTP response status codes for "List branches for HEAD commit"

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

# Code samples for "List branches for HEAD commit"



### Response

Example response Response schema

Status: 200

[ { "name": "branch\_5", "commit": { "sha": "c5b97d5ae6c19d5c5df7la34c7fbeeda2479ccbc", "url": "https://api.github.com/repos/octocat/Hello-World/commits/c5b97d5ae6c19d5c5df7la34c7fbeeda2479ccbc" }, "protected": false } ]

# List pull requests associated with a commit &

# Works with <u>GitHub Apps</u>

Lists the merged pull request that introduced the commit to the repository. If the commit is not present in the default branch, will only return open pull requests associated with the commit.

To list the open or merged pull requests associated with a branch, you can set the commit\_sha parameter to the branch name.

### Parameters for "List pull requests associated with a commit"

#### 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.

commit\_sha string Required

The SHA of the commit.

### **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 pull requests associated with a commit"

Status code Description

200 OK

# Code samples for "List pull requests associated with a commit"

GET /repos/{owner}/{repo}/commits/{commit\_sha}/pulls

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/repos/OWNER/REPO/commits/COMMIT_SHA/pulls
```

#### Response

```
Example response Response schema

Status: 200

[ { "url": "https://api.github.com/repos/octocat/Hello-World/pulls/1347", "id": 1, "node_id": "MDEx0lB1bGxSZXF1ZXN0MQ==", "html_url": "https://github.com/octocat/Hello-World/pull/1347", "diff_url": "https://github.com/octocat/Hello-World/pull/1347.patch", "issue_url": "https://api.github.com/repos/octocat/Hello-World/pulls/1347.com/repos/octocat/Hello-World/pulls/1347/comments", "review_comments_url": "https://api.github.com/repos/octocat/Hello-World/pulls/1347/comments", "review_comment_url": "https://api.github.com/repos/octocat/Hello-World/pulls/1347/comments_url": "review_comment_url": "https://api.github.com/repos/octocat/Hello-World/pulls/comments_url": "veview_comments_url": "https://api.github.com/repos/octocat/Hello-World/pulls/comments_url": "veview_comments_url": "veview_comments_u
```

# Get a commit &

# Works with <u>GitHub Apps</u>

Returns the contents of a single commit reference. You must have read access for the repository to use this endpoint.

**Note:** If there are more than 300 files in the commit diff, the response will include pagination link headers for the remaining files, up to a limit of 3000 files. Each page contains the static commit information, and the only changes are to the file listing.

You can pass the appropriate <u>media type</u> to fetch diff and patch formats. Diffs with binary data will have no patch property.

To return only the SHA-1 hash of the commit reference, you can provide the sha custom <u>media type</u> in the Accept header. You can use this endpoint to check if a remote reference's SHA-1 hash is the same as your local reference's SHA-1 hash by providing the local SHA-1 reference as the ETag.

### Signature verification object

The response will include a verification object that describes the result of verifying the commit's signature. The following fields are included in the verification object:

Name	Туре	Description
verified	boolean	Indicates whether GitHub considers the signature in this commit to be verified.
reason	string	The reason for verified value. Possible values and their meanings are enumerated in table below.
signature	string	The signature that was extracted from the commit.
payload	string	The value that was signed.

These are the possible values for reason in the verification object:

Value Description

The key that made the signature is expired.
The "signing" flag is not among the usage flags in the GPG key that made the signature.
There was an error communicating with the signature verification service.
The signature verification service is currently unavailable.
The object does not include a signature.
A non-PGP signature was found in the commit.
No user was associated with the committer email address in the commit.
The committer email address in the commit was associated with a user, but the email address is not verified on their account.
The committer email address in the commit is not included in the identities of the PGP key that made the signature.
The key that made the signature has not been registered with any user's account.
There was an error parsing the signature.
The signature could not be cryptographically verified using the key whose key-id was found in the signature.
None of the above errors applied, so the signature is considered to be verified.

# Parameters for "Get a commit"

### 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.

# **Query parameters**

page integer

Page number of the results to fetch.

Default: 1

per\_page integer

The number of results per page (max 100).

Default: 30

### HTTP response status codes for "Get a commit"

Status code	Description
200	OK
404	Resource not found
422	Validation failed, or the endpoint has been spammed.
500	Internal Error
503	Service unavailable

# Code samples for "Get a commit"



# Response

Example response Response schema

Status: 200

{ "url": "https://api.github.com/repos/octocat/Hello-World/commits/6dcb09b5b57875f334f6laebed695e2e4193db5e", "sha":

"6dcb09b5b57875f334f6laebed695e2e4193db5e", "node\_id": "MDY6Q29tbWl0NmRjYjA5YjViNTc4NzVmMzM0ZjYxYWViZWQ2OTVlMmU0MTkzZGI1ZQ==",

"html\_url": "https://github.com/octocat/Hello-World/commit/6dcb09b5b57875f334f6laebed695e2e4193db5e", "comments\_url":

"https://api.github.com/repos/octocat/Hello-World/commits/6dcb09b5b57875f334f6laebed695e2e4193db5e/comments", "commit": { "url":

"https://api.github.com/repos/octocat/Hello-World/git/commits/6dcb09b5b57875f334f6laebed695e2e4193db5e", "author": { "name":

"Monalisa Octocat", "email": "mona@github.com", "date": "2011-04-14T16:00:49Z" }, "committer": { "name": "Monalisa Octocat",

# Compare two commits @

# Works with <u>GitHub Apps</u>

Compares two commits against one another. You can compare branches in the same repository, or you can compare branches that exist in different repositories within the same repository network, including fork branches. For more information about how to view a repository's network, see "<u>Understanding connections between repositories</u>."

This endpoint is equivalent to running the <code>git log BASE..HEAD</code> command, but it returns commits in a different order. The <code>git log BASE..HEAD</code> command returns commits in reverse chronological order, whereas the API returns commits in chronological order. You can pass the appropriate <code>media type</code> to fetch diff and patch formats.

The API response includes details about the files that were changed between the two commits. This includes the status of the change (if a file was added, removed, modified, or renamed), and details of the change itself. For example, files with a renamed status have a previous\_filename field showing the previous filename of the file, and files with a modified status have a patch field showing the changes made to the file.

When calling this endpoint without any paging parameter ( per\_page or page ), the returned list is limited to 250 commits, and the last commit in the list is the most recent of the entire comparison.

# Working with large comparisons

To process a response with a large number of commits, use a query parameter ( per\_page or page ) to paginate the results. When using pagination:

- The list of changed files is only shown on the first page of results, but it includes all changed files for the entire comparison.
- The results are returned in chronological order, but the last commit in the returned list may not be the most recent one in the entire set if there are more pages of results.

For more information on working with pagination, see "Using pagination in the REST API."

# Signature verification object

The response will include a verification object that describes the result of verifying the commit's signature. The verification object includes the following fields:

Name	Туре	Description
verified	boolean	Indicates whether GitHub considers the signature in this commit to be verified.
reason	string	The reason for verified value. Possible values and their meanings are enumerated in table below.
signature	string	The signature that was extracted from the commit.
payload	string	The value that was signed.

These are the possible values for reason in the verification object:

Value	Description
expired_key	The key that made the signature is expired.
not_signing_key	The "signing" flag is not among the usage flags in the GPG key that made the signature.
<pre>gpgverify_error</pre>	There was an error communicating with the signature verification service.
gpgverify_unavailable	The signature verification service is currently unavailable.
unsigned	The object does not include a signature.
unknown_signature_type	A non-PGP signature was found in the commit.
no_user	No user was associated with the committer email address in the commit.

<pre>unverified_email</pre>	The committee email address in the commit was associated with a user, but the email address is not verified on their account.
bad_email	The committer email address in the commit is not included in the identities of the PGP key that made the signature.
unknown_key	The key that made the signature has not been registered with any user's account.
malformed_signature	There was an error parsing the signature.
invalid	The signature could not be cryptographically verified using the key whose key-id was found in the signature.
valid	None of the above errors applied, so the signature is considered to be verified.

# Parameters for "Compare two commits"

#### **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.

basehead string Required

The base branch and head branch to compare. This parameter expects the format BASE...HEAD. Both must be branch names in repo. To compare with a branch that exists in a different repository in the same network as repo, the basehead parameter expects the format USERNAME:BASE...USERNAME:HEAD.

# **Query parameters**

page integer

Page number of the results to fetch.

Default: 1

per\_page integer

The number of results per page (max 100).

Default: 30

# HTTP response status codes for "Compare two commits"

Status code Description

200 OK

404	Resource not found
500	Internal Error
503	Service unavailable

# Code samples for "Compare two commits"



### Response

Example response Response schema

Status: 200

{ "url": "https://api.github.com/repos/octocat/Hello-World/compare/master...topic", "html\_url": "https://github.com/octocat/Hello-World/compare/master...topic", "permalink\_url": "https://github.com/octocat/Hello-World/compare/octocat:bbcd538c8e72b8c175046e27cc8f907076331401...octocat:0328041d1152db8ae77652d1618a02e57f745f17", "diff\_url": "https://github.com/octocat/Hello-World/compare/master...topic.diff", "patch\_url": "https://github.com/octocat/Hello-World/compare/master...topic.diff", "patch\_url": "https://github.com/octocat/Hello-World/compare/master...topic.patch", "base\_commit": { "url": "https://api.github.com/repos/octocat/Hello-World/commits/6dcb09b5b57875f334f61aebed695e2e4193db5e", "sha": "6dcb09b5b57875f334f61aebed695e2e4193db5e", "node\_id": "

# Legal

© 2023 GitHub, Inc. <u>Terms</u> <u>Privacy</u> <u>Status</u> <u>Pricing</u> <u>Expert services</u> <u>Blog</u>