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

# Releases

Use the REST API to create, modify, and delete releases and release assets.

**Note:** These endpoints replace the endpoints to manage downloads. You can retrieve the download count and browser download URL from these endpoints.

## List releases

✓ Works with GitHub Apps

This returns a list of releases, which does not include regular Git tags that have not been associated with a release. To get a list of Git tags, use the Repository Tags API.

Information about published releases are available to everyone. Only users with push access will receive listings for draft releases.

### Parameters for "List releases"

#### 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.

#### 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 releases"

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

### Code samples for "List releases"

GET `/repos/{owner}/{repo}/releases`

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/releases
```

#### Response

Example response Response schema

```
Status: 200
```

```
 [ { "url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1", "html_url": "https://github.com/octocat/Hello-World/releases/v1.0.0",
"assets_url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1/assets", "upload_url":
"https://uploads.github.com/repos/octocat/Hello-World/releases/1/assets{?name,label}", "tarball_url":
"https://HOSTNAME/repos/octocat/Hello-World/tarball/v1.0.0", "zipball_url": "https://HOSTNAME/repos/octocat/Hello-World/zipball/v1.0.0",
"id": 1, "node_id": "MDc6UmVsZWFzZTE=", "tag_name": "v1.0.0", "target_commitish": "master", "name": "v1.0.0", "body": "Description of the
release", "draft": false, "prerelease": false, "created_at": "2013-02-27T19:35:32Z", "published_at": "2013-02-27T19:35:32Z", "author": {
```

## Create a release

Works with GitHub Apps

Users with push access to the repository can create a release.

This endpoint triggers notifications. Creating content too quickly using this endpoint may result in secondary rate limiting. See "Secondary rate limits" and "Dealing with secondary rate limits" for details.

### Parameters for "Create a release"

#### 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**

**`tag_name`** string Required

The name of the tag.

**`target_commitish`** string

Specifies the commitish value that determines where the Git tag is created from. Can be any branch or commit SHA. Unused if the Git tag already exists. Default: the repository's default branch.

**`name`** string

The name of the release.

**`body`** string

Text describing the contents of the tag.

**`draft`** boolean

`true` to create a draft (unpublished) release, `false` to create a published one.

Default: `false`

**`prerelease`** boolean

`true` to identify the release as a prerelease. `false` to identify the release as a full release.

Default: `false`

**`generate_release_notes`** boolean

Whether to automatically generate the name and body for this release. If `name` is specified, the specified name will be used; otherwise, a name will be automatically generated. If `body` is specified, the body will be pre-pended to the automatically generated notes.

Default: `false`

**`make_latest`** string

Specifies whether this release should be set as the latest release for the repository. Drafts and prereleases cannot be set as latest. Defaults to `true` for newly published releases. `legacy` specifies that the latest release should be determined based on the release creation date and higher semantic version.

Default: `true`
Can be one of: `true`, `false`, `legacy`

### HTTP response status codes for "Create a release"

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

### Code samples for "Create a release"

POST `/repos/{owner}/{repo}/releases`

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/releases \ -d
'{"tag_name":"v1.0.0","target_commitish":"master","name":"v1.0.0","body":"Description of the
release","draft":false,"prerelease":false,"generate_release_notes":false}'
```

**Response**

Example response    Response schema

Status: 201

```
{ "url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1", "html_url": "https://github.com/octocat/Hello-World/releases/v1.0.0",
"assets_url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1/assets", "upload_url":
"https://uploads.github.com/repos/octocat/Hello-World/releases/1/assets{?name,label}", "tarball_url":
"https://HOSTNAME/repos/octocat/Hello-World/tarball/v1.0.0", "zipball_url": "https://HOSTNAME/repos/octocat/Hello-World/zipball/v1.0.0",
"discussion_url": "https://github.com/octocat/Hello-World/discussions/90", "id": 1, "node_id": "MDc6UmVsZWFzZTE=", "tag_name": "v1.0.0",
"target_commitish": "master", "name": "v1.0.0", "body": "Description of the release", "draft": false, "prerelease": false, "created_at":
```

## Generate release notes content for a release

✔ Works with GitHub Apps

Generate a name and body describing a release. The body content will be markdown formatted and contain information like the changes since last release and users who contributed. The generated release notes are not saved anywhere. They are intended to be generated and used when creating a new release.

### Parameters for "Generate release notes content for a release"

#### 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

`tag_name` string Required

The tag name for the release. This can be an existing tag or a new one.

`target_commitish` string

Specifies the commitish value that will be the target for the release's tag. Required if the supplied tag_name does not reference an existing tag. Ignored if the tag_name already exists.

`previous_tag_name` string

The name of the previous tag to use as the starting point for the release notes. Use to manually specify the range for the set of changes considered as part this release.

`configuration_file_path` string

Specifies a path to a file in the repository containing configuration settings used for generating the release notes. If unspecified, the configuration file located in the repository at '.github/release.yml' or '.github/release.yaml' will be used. If that is not present, the default configuration will be used.

### HTTP response status codes for "Generate release notes content for a release"

| Status code | Description |
|---|---|
| `200` | Name and body of generated release notes |
| `404` | Resource not found |

### Code samples for "Generate release notes content for a release"

POST /repos/{owner}/{repo}/releases/generate-notes

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/releases/generate-notes \ -d '{"tag_name":"v1.0.0","target_commitish":"main","previous_tag_name":"v0.9.2","configuration_file_path":".github/custom_release_config.yml"}'
```

### Name and body of generated release notes

Example response Response schema

Status: 200

```
{ "name": "Release v1.0.0 is now available!", "body": "##Changes in Release v1.0.0 ... ##Contributors @monalisa" }
```

## Get the latest release

Works with [GitHub Apps]

View the latest published full release for the repository.

The latest release is the most recent non-prerelease, non-draft release, sorted by the `created_at` attribute. The `created_at` attribute is the date of the commit used for the release, and not the date when the release was drafted or published.

### Parameters for "Get the latest release"

#### 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.

### HTTP response status codes for "Get the latest release"

| Status code | Description |
|---|---|
| `200` | OK |

### Code samples for "Get the latest release"

GET /repos/{owner}/{repo}/releases/latest

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/releases/latest
```

#### Response

Example response Response schema

Status: 200

```
{ "url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1", "html_url": "https://github.com/octocat/Hello-World/releases/v1.0.0",
"assets_url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1/assets", "upload_url":
"https://uploads.github.com/repos/octocat/Hello-World/releases/1/assets{?name,label}", "tarball_url":
"https://HOSTNAME/repos/octocat/Hello-World/tarball/v1.0.0", "zipball_url": "https://HOSTNAME/repos/octocat/Hello-World/zipball/v1.0.0",
"discussion_url": "https://github.com/octocat/Hello-World/discussions/90", "id": 1, "node_id": "MDc6UmVsZWFzZTE=", "tag_name": "v1.0.0",
"target_commitish": "master", "name": "v1.0.0", "body": "Description of the release", "draft": false, "prerelease": false, "created_at":
```

## Get a release by tag name

✓ Works with GitHub Apps

Get a published release with the specified tag.

### Parameters for "Get a release by tag name"

#### 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.

`tag` string Required

tag parameter

### HTTP response status codes for "Get a release by tag name"

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

### Code samples for "Get a release by tag name"

GET /repos/{owner}/{repo}/releases/tags/{tag}

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/releases/tags/TAG
```

#### Response

Example response Response schema

```
Status: 200
```

```
 { "url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1", "html_url": "https://github.com/octocat/Hello-World/releases/v1.0.0",
"assets_url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1/assets", "upload_url":
"https://uploads.github.com/repos/octocat/Hello-World/releases/1/assets{?name,label}", "tarball_url":
"https://HOSTNAME/repos/octocat/Hello-World/tarball/v1.0.0", "zipball_url": "https://HOSTNAME/repos/octocat/Hello-World/zipball/v1.0.0",
"discussion_url": "https://github.com/octocat/Hello-World/discussions/90", "id": 1, "node_id": "MDc6UmVsZWFzZTE=", "tag_name": "v1.0.0",
"target_commitish": "master", "name": "v1.0.0", "body": "Description of the release", "draft": false, "prerelease": false, "created_at":
```

## Get a release

✓ Works with GitHub Apps

**Note:** This returns an `upload_url` key corresponding to the endpoint for uploading release assets. This key is a hypermedia resource.

### Parameters for "Get a release"

#### 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.

**`release_id`** integer Required

The unique identifier of the release.

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

| Status code | Description |
|---|---|
| `200` | **Note:** This returns an `upload_url` key corresponding to the endpoint for uploading release assets. This key is a hypermedia resource. |
| `401` | Unauthorized |

### Code samples for "Get a release"

GET /repos/{owner}/{repo}/releases/{release_id}

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/releases/RELEASE_ID
```

**Note: This returns an `upload_url` key corresponding to the endpoint for uploading release assets. This key is a hypermedia resource.**

Example response Response schema

Status: 200

```
{ "url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1", "html_url": "https://github.com/octocat/Hello-World/releases/v1.0.0",
"assets_url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1/assets", "upload_url":
"https://uploads.github.com/repos/octocat/Hello-World/releases/1/assets{?name,label}", "tarball_url":
"https://HOSTNAME/repos/octocat/Hello-World/tarball/v1.0.0", "zipball_url": "https://HOSTNAME/repos/octocat/Hello-World/zipball/v1.0.0",
"discussion_url": "https://github.com/octocat/Hello-World/discussions/90", "id": 1, "node_id": "MDc6UmVsZWFzZTE=", "tag_name": "v1.0.0",
"target_commitish": "master", "name": "v1.0.0", "body": "Description of the release", "draft": false, "prerelease": false, "created_at":
```

## Update a release

Works with GitHub Apps

Users with push access to the repository can edit a release.

### Parameters for "Update a release"

#### 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.

`release_id` integer Required

The unique identifier of the release.

**Body parameters**

`tag_name` string

The name of the tag.

`target_commitish` string

Specifies the commitish value that determines where the Git tag is created from. Can be any branch or commit SHA. Unused if the Git tag already exists. Default: the repository's default branch.

`name` string

The name of the release.

`body` string

Text describing the contents of the tag.

`draft` boolean

`true` makes the release a draft, and `false` publishes the release.

`prerelease` boolean

`true` to identify the release as a prerelease, `false` to identify the release as a full release.

`make_latest` string

Specifies whether this release should be set as the latest release for the repository. Drafts and prereleases cannot be set as latest. Defaults to `true` for newly published releases. `legacy` specifies that the latest release should be determined based on the release creation date and higher semantic version.

Default: `true`
Can be one of: `true`, `false`, `legacy`

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

| Status code | Description |
|---|---|
| `200` | OK |

### Code samples for "Update a release"

PATCH /repos/{owner}/{repo}/releases/{release_id}

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/releases/RELEASE_ID \ -d
'{"tag_name":"v1.0.0","target_commitish":"master","name":"v1.0.0","body":"Description of the release","draft":false,"prerelease":false}'
```

**Response**

Example response Response schema

```
Status: 200
```

```
{ "url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1", "html_url": "https://github.com/octocat/Hello-World/releases/v1.0.0",
"assets_url": "https://HOSTNAME/repos/octocat/Hello-World/releases/1/assets", "upload_url":
"https://uploads.github.com/repos/octocat/Hello-World/releases/1/assets{?name,label}", "tarball_url":
"https://HOSTNAME/repos/octocat/Hello-World/tarball/v1.0.0", "zipball_url": "https://HOSTNAME/repos/octocat/Hello-World/zipball/v1.0.0",
"discussion_url": "https://github.com/octocat/Hello-World/discussions/90", "id": 1, "node_id": "MDc6UmVsZWFzZTE=", "tag_name": "v1.0.0",
"target_commitish": "master", "name": "v1.0.0", "body": "Description of the release", "draft": false, "prerelease": false, "created_at":
```

## Delete a release

✓ Works with GitHub Apps

Users with push access to the repository can delete a release.

### Parameters for "Delete a release"

#### 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.

`release_id` integer Required

The unique identifier of the release.

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

| Status code | Description |
|---|---|
| `204` | No Content |

### Code samples for "Delete a release"

DELETE /repos/{owner}/{repo}/releases/{release_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" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/releases/RELEASE_ID
```

#### Response

```
Status: 204
```

### Legal