

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

Repository webhooks

Use the REST API to create and manage webhooks for your repositories.

About repository webhooks

Repository webhooks allow your server to receive HTTP `POST` payloads whenever certain events happen in a repository. For more information, see "[Webhooks documentation](#)."

List repository webhooks

 Works with [GitHub Apps](#)

Lists webhooks for a repository. `last_response` may return null if there have not been any deliveries within 30 days.

Parameters for "List repository webhooks"

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 repository webhooks"

Status code	Description
200	OK
404	Resource not found

Code samples for "List repository webhooks"

GET

/repos/{owner}/{repo}/hooks

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/hooks
```

Response

Example response

Response schema

Status: 200

```
[ { "type": "Repository", "id": 12345678, "name": "web", "active": true, "events": [ "push", "pull_request" ], "config": { "content_type": "json", "insecure_ssl": "0", "url": "https://example.com/webhook" }, "updated_at": "2019-06-03T00:57:16Z", "created_at": "2019-06-03T00:57:16Z", "url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678", "test_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/test", "ping_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/pings", "deliveries_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/deliveries", "last_response": { "code": null, "status": "unused", "message": null } } ]
```

Create a repository webhook

✔ Works with [GitHub Apps](#)

Repositories can have multiple webhooks installed. Each webhook should have a unique `config`. Multiple webhooks can share the same `config` as long as those webhooks do not have any `events` that overlap.

Parameters for "Create a repository webhook"

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

name string

Use `web` to create a webhook. Default: `web`. This parameter only accepts the value `web`.

config object

Key/value pairs to provide settings for this webhook.

► Properties of `config`

events array of strings

Determines what [events](#) the hook is triggered for.

Default: `["push"]`

active boolean

Determines if notifications are sent when the webhook is triggered. Set to `true` to send notifications.

Default: `true`

HTTP response status codes for "Create a repository webhook"

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 repository webhook"

POST

/repos/{owner}/{repo}/hooks

cURL

JavaScript



```
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/hooks \ -d '{"name":"web","active":true,"events":["push","pull_request"],"config":{"url":"https://example.com/webhook","content_type":"json","insecure_ssl":"0"}}'
```

Response

Example response

Response schema

Status: 201

```
{ "type": "Repository", "id": 12345678, "name": "web", "active": true, "events": [ "push", "pull_request" ], "config": { "content_type": "json", "insecure_ssl": "0", "url": "https://example.com/webhook" }, "updated_at": "2019-06-03T00:57:16Z", "created_at": "2019-06-03T00:57:16Z", "url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678", "test_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/test", "ping_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/pings", "deliveries_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/deliveries",
```

```
"last_response": { "code": null, "status": "unused", "message": null } }
```

Get a repository webhook

✔ Works with [GitHub Apps](#)

Returns a webhook configured in a repository. To get only the webhook `config` properties, see "[Get a webhook configuration for a repository](#)."

Parameters for "Get a repository webhook"

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.

`hook_id` integer **Required**

The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

HTTP response status codes for "Get a repository webhook"

Status code	Description
200	OK
404	Resource not found

Code samples for "Get a repository webhook"

GET

/repos/{owner}/{repo}/hooks/{hook_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" \ https://api.github.com/repos/OWNER/REPO/hooks/HOOK_ID
```

Response

Example response

Response schema

Status: 200

```
{ "type": "Repository", "id": 12345678, "name": "web", "active": true, "events": [ "push", "pull_request" ], "config": {
"content_type": "json", "insecure_ssl": "0", "url": "https://example.com/webhook" }, "updated_at": "2019-06-03T00:57:16Z",
"created_at": "2019-06-03T00:57:16Z", "url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678", "test_url":
"https://api.github.com/repos/octocat/Hello-World/hooks/12345678/test", "ping_url": "https://api.github.com/repos/octocat/Hello-
World/hooks/12345678/pings", "deliveries_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/deliveries",
"last_response": { "code": null, "status": "unused", "message": null } }
```

Update a repository webhook

✔ Works with [GitHub Apps](#)

Updates a webhook configured in a repository. If you previously had a `secret` set, you must provide the same `secret` or set a new `secret` or the secret will be removed. If you are only updating individual webhook `config` properties, use "[Update a webhook configuration for a repository](#)."

Parameters for "Update a repository webhook"

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.

hook_id integer **Required**
The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

Body parameters

config object
Key/value pairs to provide settings for this webhook.

► Properties of `config`

events array of strings
Determines what [events](#) the hook is triggered for. This replaces the entire array of events.
Default: `["push"]`

add_events array of strings
Determines a list of events to be added to the list of events that the Hook triggers for.

Determines a list of events to be added to the list of events that the Hook triggers for.

remove_events array of strings

Determines a list of events to be removed from the list of events that the Hook triggers for.

active boolean

Determines if notifications are sent when the webhook is triggered. Set to `true` to send notifications.

Default: `true`

HTTP response status codes for "Update a repository webhook"

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

Code samples for "Update a repository webhook"

PATCH

`/repos/{owner}/{repo}/hooks/{hook_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" \
https://api.github.com/repos/OWNER/REPO/hooks/HOOK_ID \
-d '{"active":true,"add_events":["pull_request"]}'
```

Response

Example response

Response schema

Status: 200

```
{ "type": "Repository", "id": 12345678, "name": "web", "active": true, "events": [ "push", "pull_request" ], "config": { "content_type": "json", "insecure_ssl": "0", "url": "https://example.com/webhook" }, "updated_at": "2019-06-03T00:57:16Z", "created_at": "2019-06-03T00:57:16Z", "url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678", "test_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/test", "ping_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/pings", "deliveries_url": "https://api.github.com/repos/octocat/Hello-World/hooks/12345678/deliveries", "last_response": { "code": null, "status": "unused", "message": null } }
```

Delete a repository webhook

✔ Works with [GitHub Apps](#)

Parameters for "Delete a repository webhook"

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.

hook_id integer **Required**

The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

HTTP response status codes for "Delete a repository webhook"

Status code	Description
204	No Content
404	Resource not found

Code samples for "Delete a repository webhook"

DELETE /repos/{owner}/{repo}/hooks/{hook_id}

cURLJavaScriptGitHub 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/repos/OWNER/REPO/hooks/HOOK_ID
```

Response

Status: 204

Get a webhook configuration for a repository

✔ Works with [GitHub Apps](#)

Returns the webhook configuration for a repository. To get more information about the webhook, including the `active` state and `events`, use "[Get a repository webhook](#)."

Access tokens must have the `read:repo_hook` or `repo` scope, and GitHub Apps must have the `repository_hooks:read` permission.

Parameters for "Get a webhook configuration for a repository"

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.

hook_id integer **Required**

The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

HTTP response status codes for "Get a webhook configuration for a repository"

Status code	Description
200	OK

Code samples for "Get a webhook configuration for a repository"

GET /repos/{owner}/{repo}/hooks/{hook_id}/config

cURLJavaScriptGitHub 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/hooks/HOOK_ID/config
```

Response

Example responseResponse schema

Status: 200

```
{ "content_type": "json", "insecure_ssl": "0", "secret": "*****", "url": "https://example.com/webhook" }
```

Update a webhook configuration for a repository

✔ Works with [GitHub Apps](#)

Updates the webhook configuration for a repository. To update more information about the webhook, including the

active state and events , use "[Update a repository webhook](#)."

Access tokens must have the write:repo_hook or repo scope, and GitHub Apps must have the repository_hooks:write permission.

Parameters for "Update a webhook configuration for a repository"

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.

hook_id integer Required
The unique identifier of the hook. You can find this value in the X-GitHub-Hook-ID header of a webhook delivery.

Body parameters

url string
The URL to which the payloads will be delivered.

content_type string
The media type used to serialize the payloads. Supported values include json and form . The default is form .

secret string
If provided, the secret will be used as the key to generate the HMAC hex digest value for [delivery signature headers](#).

insecure_ssl string or number
Determines whether the SSL certificate of the host for url will be verified when delivering payloads. Supported values include 0 (verification is performed) and 1 (verification is not performed). The default is 0 . **We strongly recommend not setting this to 1 as you are subject to man-in-the-middle and other attacks.**

HTTP response status codes for "Update a webhook configuration for a repository"

Status code	Description
200	OK

Code samples for "Update a webhook configuration for a repository"

PATCH /repos/{owner}/{repo}/hooks/{hook_id}/config

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/repos/OWNER/REPO/hooks/HOOK_ID/config \ -d
```

```
'{"content_type":"json","url":"https://example.com/webhook"}'
```

Response

Example response

Response schema

Status: 200

```
{ "content_type": "json", "insecure_ssl": "0", "secret": "*****", "url": "https://example.com/webhook" }
```

List deliveries for a repository webhook

✔ Works with [GitHub Apps](#)

Returns a list of webhook deliveries for a webhook configured in a repository.

Parameters for "List deliveries for a repository webhook"

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.

hook_id integer **Required**

The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

Query parameters

per_page integer

The number of results per page (max 100).

Default: `30`

cursor string

Used for pagination: the starting delivery from which the page of deliveries is fetched. Refer to the **link** header for the next and previous page cursors.

redelivery boolean

HTTP response status codes for "List deliveries for a repository webhook"

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

Code samples for "List deliveries for a repository webhook"

GET

/repos/{owner}/{repo}/hooks/{hook_id}/deliveries

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/hooks/HOOK_ID/deliveries
```

Response

Example response

Response schema

Status: 200

```
[ { "id": 12345678, "guid": "0b989ba4-242f-11e5-81e1-c7b6966d2516", "delivered_at": "2019-06-03T00:57:16Z", "redelivery": false, "duration": 0.27, "status": "OK", "status_code": 200, "event": "issues", "action": "opened", "installation_id": 123, "repository_id": 456 }, { "id": 123456789, "guid": "0b989ba4-242f-11e5-81e1-c7b6966d2516", "delivered_at": "2019-06-04T00:57:16Z", "redelivery": true, "duration": 0.28, "status": "OK", "status_code": 200, "event": "issues", "action": "opened", "installation_id": 123, "repository_id": 456 } ]
```

Get a delivery for a repository webhook

✔ Works with [GitHub Apps](#)

Returns a delivery for a webhook configured in a repository.

Parameters for "Get a delivery for a repository webhook"

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.

hook_id integer **Required**

The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

delivery_id integer **Required**

HTTP response status codes for "Get a delivery for a repository webhook"

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

Code samples for "Get a delivery for a repository webhook"

GET `/repos/{owner}/{repo}/hooks/{hook_id}/deliveries/{delivery_id}`

cURLJavaScriptGitHub 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/hooks/HOOK_ID/deliveries/DELIVERY_ID
```

Response

Example responseResponse schema

Status: 200

```
{ "id": 12345678, "guid": "0b989ba4-242f-11e5-81e1-c7b6966d2516", "delivered_at": "2019-06-03T00:57:16Z", "redelivery": false, "duration": 0.27, "status": "OK", "status_code": 200, "event": "issues", "action": "opened", "installation_id": 123, "repository_id": 456, "url": "https://www.example.com", "request": { "headers": { "X-GitHub-Delivery": "0b989ba4-242f-11e5-81e1-c7b6966d2516", "X-Hub-Signature-256": "sha256=6dc09b5b57875f334f61aebd695e2e4193db5e", "Accept": "*/*", "X-GitHub-Hook-ID": "42", "User-Agent": "GitHub-Hookshot/b8c71d8", "X-GitHub-Event": "issues", "X-GitHub-Hook-Installation-Target-ID": "123", "X-GitHub-Hook-Installation-Target-Type": "repository", "content-type": "application/json", "X-Hub-Signature":
```

Redeliver a delivery for a repository webhook

Redeliver a webhook delivery for a webhook configured in a repository.

Parameters for "Redeliver a delivery for a repository webhook"

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.

hook_id integer **Required**

The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

delivery_id integer **Required**

HTTP response status codes for "Redeliver a delivery for a repository webhook"

Status code	Description
202	Accepted
400	Bad Request
422	Validation failed, or the endpoint has been spammed.

Code samples for "Redeliver a delivery for a repository webhook"

POST /repos/{owner}/{repo}/hooks/{hook_id}/deliveries/{delivery_id}/attempts

cURLJavaScriptGitHub 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/repos/OWNER/REPO/hooks/HOOK_ID/deliveries/DELIVERY_ID/attempts
```

Accepted

Example response

Response schema

Status: 202

Ping a repository webhook

✔ Works with [GitHub Apps](#)

This will trigger a [ping event](#) to be sent to the hook.

Parameters for "Ping a repository webhook"

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.

hook_id integer **Required**

The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

HTTP response status codes for "Ping a repository webhook"

Status code	Description
204	No Content
404	Resource not found

Code samples for "Ping a repository webhook"

POST

`/repos/{owner}/{repo}/hooks/{hook_id}/pings`

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/repos/OWNER/REPO/hooks/HOOK_ID/pings
```

Response

Status: 204

Test the push repository webhook [↗](#)

✔ Works with [GitHub Apps](#)

This will trigger the hook with the latest push to the current repository if the hook is subscribed to `push` events. If the hook is not subscribed to `push` events, the server will respond with 204 but no test POST will be generated.

Note: Previously `/repos/:owner/:repo/hooks/:hook_id/test`

Parameters for "Test the push repository webhook"

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.

hook_id integer **Required**
The unique identifier of the hook. You can find this value in the `X-GitHub-Hook-ID` header of a webhook delivery.

HTTP response status codes for "Test the push repository webhook"

Status code	Description
204	No Content
404	Resource not found

Code samples for "Test the push repository webhook"

POST `/repos/{owner}/{repo}/hooks/{hook_id}/tests`

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/repos/OWNER/REPO/hooks/HOOK_ID/tests
```

Response

Status: 204

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)