# Webhook events and payloads

**In this article**

About webhook events and payloads
branch_protection_configuration
branch_protection_rule
check_run
check_suite
code_scanning_alert
commit_comment
create
delete
dependabot_alert
deploy_key
deployment
deployment_protection_rule
deployment_review
deployment_status
discussion
discussion_comment
fork
github_app_authorization
gollum
installation
installation_repositories
installation_target
issue_comment
issues
label
marketplace_purchase
member
membership
merge_group
meta
milestone
org_block
organization
package
page_build
personal_access_token_request
ping
project_card
project
project_column
projects_v2
projects_v2_item

- public
- pull_request
- pull_request_review_comment
- pull_request_review
- pull_request_review_thread
- push
- registry_package
- release
- repository_advisory
- repository
- repository_dispatch
- repository_import
- repository_ruleset
- repository_vulnerability_alert
- secret_scanning_alert
- secret_scanning_alert_location
- security_advisory
- security_and_analysis
- sponsorship
- star
- status
- team_add
- team
- user
- watch
- workflow_dispatch
- workflow_job
- workflow_run

---

Learn about when each webhook event occurs and what the payload contains.

## About webhook events and payloads

You can create webhooks that subscribe to the events listed on this page. To limit the number of HTTP requests to your server, you should only subscribe to the specific events that you plan on handling. For more information, see "Creating webhooks."

Each webhook event on this page includes a description of the webhook properties for that event. If the event has multiple actions, the properties corresponding to each action are included.

Each event is only available to specific types of webhooks. For example, an organization webhook can subscribe to the `team` event, but a repository webhook cannot. The description of each webhook event lists the availability for that event. For more information, see "Types of webhooks."

### Payload cap

Payloads are capped at 25 MB. If an event generates a larger payload, GitHub will not deliver a payload for that webhook event. This may happen, for example, on a `create`

event if many branches or tags are pushed at once. We suggest monitoring your payload size to ensure delivery.

## Delivery headers

HTTP POST payloads that are delivered to your webhook's configured URL endpoint will contain several special headers:

- `X-GitHub-Hook-ID`: The unique identifier of the webhook.
- `X-GitHub-Event`: The name of the event that triggered the delivery.
- `X-GitHub-Delivery`: A globally unique identifier (GUID) to identify the delivery.
- `X-Hub-Signature`: This header is sent if the webhook is configured with a `secret`. This is the HMAC hex digest of the request body, and is generated using the SHA-1 hash function and the `secret` as the HMAC `key`. `X-Hub-Signature` is provided for compatibility with existing integrations. We recommend that you use the more secure `X-Hub-Signature-256` instead.
- `X-Hub-Signature-256`: This header is sent if the webhook is configured with a `secret`. This is the HMAC hex digest of the request body, and is generated using the SHA-256 hash function and the `secret` as the HMAC `key`. For more information, see "Validating webhook deliveries."
- `User-Agent`: This header will always have the prefix `GitHub-Hookshot/`.
- `X-GitHub-Hook-Installation-Target-Type`: The type of resource where the webhook was created.
- `X-GitHub-Hook-Installation-Target-ID`: The unique identifier of the resource where the webhook was created.

To see what each header might look like in a webhook payload, see "Example webhook delivery."

## Example webhook delivery

You can choose to have payloads delivered in JSON format ( `application/json` ) or as URL-encoded data ( `x-www-form-urlencoded` ). Following is an example of a webhook POST request that uses the JSON format.

```
> POST /payload HTTP/2

> X-GitHub-Delivery: 72d3162e-cc78-11e3-81ab-4c9367dc0958
> X-Hub-Signature: sha1=7d38cdd689735b008b3c702edd92eea23791c5f6
> X-Hub-Signature-256:
sha256=d57c68ca6f92289e6987922ff26938930f6e66a2d161ef06abdf1859230aa23c
> User-Agent: GitHub-Hookshot/044aadd
> Content-Type: application/json
> Content-Length: 6615
> X-GitHub-Event: issues
> X-GitHub-Hook-ID: 292430182
> X-GitHub-Hook-Installation-Target-ID: 79929171
> X-GitHub-Hook-Installation-Target-Type: repository

> {
>   "action": "opened",
>   "issue": {
>     "url": "https://api.github.com/repos/octocat/Hello-World/issues/1347",
>     "number": 1347,
>     ...
>   },
>   "repository" : {
>     "id": 1296269,
>     "full_name": "octocat/Hello-World",
>     "owner": {
>       "login": "octocat",
>       "id": 1,
```

```
>       ...
>     },
>     ...
>   },
>   "sender": {
>     "login": "octocat",
>     "id": 1,
>     ...
>   }
> }
```

## branch_protection_configuration

This event occurs when there is a change to branch protection configurations for a repository. For more information, see "About protected branches." For information about using the APIs to manage branch protection rules, see "Branch protection rule" in the GraphQL documentation or "Branch protection" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Administration" repository permission.

### Availability for `branch_protection_configuration`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `branch_protection_configuration`

Action type: **disabled**

All branch protections were disabled for a repository.

#### Webhook request body parameters

**`action`** string Required

Value: `disabled`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository`

property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## branch_protection_rule

This event occurs when there is activity relating to branch protection rules. For more information, see "About protected branches." For information about the APIs to manage branch protection rules, see the GraphQL documentation or "Branch protection" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Administration" repository permission.

### Availability for `branch_protection_rule`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `branch_protection_rule`

Action type: **created**

A branch protection rule was created.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`rule`** object Required

The branch protection rule. Includes a `name` and all the branch protection settings applied to

branches that match the name. Binary settings are boolean. Multi-level configurations are one of `off`, `non_admins`, or `everyone`. Actor and build lists are arrays of strings.

▶ Properties of `rule`

---

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## check_run

This event occurs when there is activity relating to a check run. For information about check runs, see "Getting started with the Checks API." For information about the APIs to manage check runs, see the GraphQL API documentation or "Check Runs" in the REST API documentation.

For activity relating to check suites, use the `check-suite` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Checks" repository permission. To receive the `rerequested` and `requested_action` event types, the app must have at least write-level access for the "Checks" permission. GitHub Apps with write-level access for the "Checks" permission are automatically subscribed to this webhook event.

Repository and organization webhooks only receive payloads for the `created` and `completed` event types in repositories.

**Note**: The API only looks for pushes in the repository where the check run was created. Pushes to a branch in a forked repository are not detected and return an empty `pull_requests` array and a `null` value for `head_branch`.

### Availability for `check_run`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `check_run`

Action type: **completed**

A check run was completed, and a conclusion is available.

#### Webhook request body parameters

---

**`action`** string

Value: `completed`

---

**`check_run`** object Required

A check performed on the code of a given code change

▶ Properties of `check_run`

---

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

`organization` object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

`repository` object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

`sender` object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## check_suite

This event occurs when there is activity relating to a check suite. For information about check suites, see "Getting started with the Checks API." For information about the APIs to manage check suites, see the GraphQL API documentation or "Check Suites" in the REST API documentation.

For activity relating to check runs, use the `check_run` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Checks" permission. To receive the `requested` and `rerequested` event types, the app must have at least write-level access for the "Checks" permission. GitHub Apps with write-level access for the "Checks" permission are automatically subscribed to this webhook event.

Repository and organization webhooks only receive payloads for the `completed` event types in repositories.

**Note**: The API only looks for pushes in the repository where the check suite was created. Pushes to a branch in a forked repository are not detected and return an empty `pull_requests` array and a `null` value for `head_branch`.

### Availability for `check_suite`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `check_suite`

Action type: **completed**

All check runs in a check suite have completed, and a conclusion is available.

#### Webhook request body parameters

`action` string Required

Value: `completed`

`check_suite` object Required

The check_suite.

▶ Properties of `check_suite`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## code_scanning_alert

This event occurs when there is activity relating to code scanning alerts in a repository. For more information, see "About code scanning" and "About code scanning alerts." For information about the API to manage code scanning, see "Code scanning" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Code scanning alerts" repository permission.

### Availability for `code_scanning_alert`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `code_scanning_alert`

Action type: **appeared_in_branch**

A previously created code scanning alert appeared in another branch. This can happen when a branch is merged into or created from a branch with a pre-existing code scanning alert.

#### Webhook request body parameters

**`action`** string Required

Value: `appeared_in_branch`

**`alert`** object Required

The code scanning alert involved in the event.

▶ Properties of `alert`

---

**`commit_oid`** string Required

The commit SHA of the code scanning alert. When the action is `reopened_by_user` or `closed_by_user`, the event was triggered by the `sender` and this value will be empty.

---

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

---

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

---

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`ref`** string Required

The Git reference of the code scanning alert. When the action is `reopened_by_user` or `closed_by_user`, the event was triggered by the `sender` and this value will be empty.

---

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## commit_comment

This event occurs when there is activity relating to commit comments. For more information about commit comments, see "Commenting on a pull request." For information about the APIs to manage commit comments, see the GraphQL API documentation or "Commit comments" in the REST API documentation.

For activity relating to comments on pull request reviews, use the `pull_request_review_comment` event. For activity relating to issue comments, use the `issue_comment` event. For activity relating to discussion comments, use the `discussion_comment` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Contents" repository permission.

### Availability for `commit_comment`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `commit_comment`

Someone commented on a commit.

#### Webhook request body parameters

**`action`** string Required

The action performed. Can be `created`.

Value: `created`

**`comment`** object Required

The commit comment resource.

▶ Properties of `comment`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# create

This event occurs when a Git branch or tag is created.

To subscribe to this event, a GitHub App must have at least read-level access for the "Contents" repository permission.

**Notes**:

- This event will not occur when more than three tags are created at once.
- Payloads are capped at 25 MB. If an event generates a larger payload, GitHub will not deliver a payload for that webhook event. This may happen, for example, if many branches or tags are pushed at once. We suggest monitoring your payload size to ensure delivery.

## Availability for `create`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `create`

#### Webhook request body parameters

**`description`** string or null Required

The repository's current description.

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`master_branch`** string Required

The name of the repository's default branch (usually `main`).

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`pusher_type`** string Required

The pusher type for the event. Can be either `user` or a deploy key.

**`ref`** string Required

The `git ref` resource.

**`ref_type`** string Required

The type of Git ref object created in the repository.

Can be one of: `tag`, `branch`

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## delete

This event occurs when a Git branch or tag is deleted. To subscribe to all pushes to a repository, including branch and tag deletions, use the `push` webhook event.

To subscribe to this event, a GitHub App must have at least read-level access for the

"Contents" repository permission.

**Note**: This event will not occur when more than three tags are deleted at once.

### Availability for `delete`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `delete`

#### Webhook request body parameters

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`pusher_type`** string Required

The pusher type for the event. Can be either `user` or a deploy key.

**`ref`** string Required

The `git ref` resource.

**`ref_type`** string Required

The type of Git ref object deleted in the repository.

Can be one of: `tag`, `branch`

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## dependabot_alert

This event occurs when there is activity relating to Dependabot alerts.

For more information about Dependabot alerts, see "About Dependabot alerts." For information about the API to manage Dependabot alerts, see "Dependabot alerts" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Dependabot alerts" repository permission.

**Note**: Webhook events for Dependabot alerts are currently in beta and subject to change.

## Availability for `dependabot_alert`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `dependabot_alert`

Action type: **auto_dismissed**

A Dependabot alert was automatically closed.

#### Webhook request body parameters

**`action`** string Required

Value: `auto_dismissed`

**`alert`** object Required

A Dependabot alert.

▶ Properties of `alert`

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# deploy_key

This event occurs when there is activity relating to deploy keys. For more information,

see "Managing deploy keys." For information about the APIs to manage deploy keys, see the GraphQL API documentation or "Deploy keys" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Deployments" repository permission.

### Availability for `deploy_key`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `deploy_key`

Action type: **created**

A deploy key was created.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`key`** object Required

The `deploy_key` resource.

▶ Properties of `key`

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## deployment

This event occurs when there is activity relating to deployments. For more information,

see "About deployments." For information about the APIs to manage deployments, see the GraphQL API documentation or "Deployments" in the REST API documentation.

For activity relating to deployment status, use the `deployment_status` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Deployments" repository permission.

## Availability for `deployment`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `deployment`

A deployment was created.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`deployment`** object Required

The deployment.

▶ Properties of `deployment`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`workflow`** object or null Required

▶ Properties of `workflow`

**`workflow_run`** object or null Required

▶ Properties of `workflow_run`

## deployment_protection_rule

This event occurs when there is activity relating to deployment protection rules. For more information, see "Using environments for deployment." For information about the API to manage deployment protection rules, see the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Deployments" repository permission.

### Availability for `deployment_protection_rule`

- GitHub Apps

### Webhook payload object for `deployment_protection_rule`

A deployment protection rule was requested for an environment.

#### Webhook request body parameters

**`action`** string

Value: `requested`

**`environment`** string

The name of the environment that has the deployment protection rule.

**`event`** string

The event that triggered the deployment protection rule.

**`deployment_callback_url`** string

The URL to review the deployment protection rule.

**`deployment`** object

A request for a specific ref(branch,sha,tag) to be deployed

▶ Properties of `deployment`

**`pull_requests`** array of objects

▶ Properties of `pull_requests`

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is

configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## deployment_review

This event occurs when there is activity relating to deployment reviews. For more information, see "About deployments." For information about the APIs to manage deployments, see the GraphQL API documentation or "Deployments" in the REST API documentation.

For activity relating to deployment creation or deployment status, use the `deployment` or `deployment_status` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Deployments" repository permission.

### Availability for `deployment_review`

- GitHub Apps

### Webhook payload object for `deployment_review`

Action type: **approved**

A deployment review was approved.

#### Webhook request body parameters

**`action`** string Required

Value: `approved`

**`approver`** object

▶ Properties of `approver`

**`comment`** string

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object Required

**`organization`** object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`reviewers`** array of objects

▶ Properties of `reviewers`

---

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

---

**`since`** string Required

---

**`workflow_job_run`** object

▶ Properties of `workflow_job_run`

---

**`workflow_job_runs`** array of objects

▶ Properties of `workflow_job_runs`

---

**`workflow_run`** object or null Required

▶ Properties of `workflow_run`

## deployment_status

This event occurs when there is activity relating to deployment statuses. For more information, see "About deployments." For information about the APIs to manage deployments, see the GraphQL API documentation or "Deployments" in the REST API documentation.

For activity relating to deployment creation, use the `deployment` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Deployments" repository permission.

### Availability for `deployment_status`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `deployment_status`

A new deployment status was created.

#### Webhook request body parameters

---

**`action`** string Required

**`action`** string Required

Value: `created`

---

**`check_run`** object or null

▶ Properties of `check_run`

---

**`deployment`** object Required

The deployment.

▶ Properties of `deployment`

---

**`deployment_status`** object Required

The deployment status.

▶ Properties of `deployment_status`

---

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

---

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

---

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

---

**`workflow`** object or null

▶ Properties of `workflow`

---

**`workflow_run`** object or null

▶ Properties of `workflow_run`

## discussion

This event occurs when there is activity relating to a discussion. For more information about discussions, see "GitHub Discussions." For information about the API to manage discussions, see the GraphQL documentation.

For activity relating to a comment on a discussion, use the `discussion_comment` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Discussions" repository permission.

**Note**: Webhook events for GitHub Discussions are currently in beta and subject to change.

## Availability for `discussion`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `discussion`

Action type: **answered**

A comment on the discussion was marked as the answer.

#### Webhook request body parameters

**`action`** string Required

Value: `answered`

**`answer`** object Required

▶ Properties of `answer`

**`discussion`** object Required

A Discussion in a repository.

▶ Properties of `discussion`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# discussion_comment

This event occurs when there is activity relating to a comment on a discussion. For more information about discussions, see "GitHub Discussions." For information about the API to manage discussions, see the GraphQL documentation.

For activity relating to a discussion as opposed to comments on a discussion, use the `discussion` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Discussions" repository permission.

**Note**: Webhook events for GitHub Discussions are currently in beta and subject to change.

## Availability for `discussion_comment`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `discussion_comment`

Action type: **created**

A comment on a discussion was created.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`comment`** object Required

▶ Properties of `comment`

**`discussion`** object Required

A Discussion in a repository.

▶ Properties of `discussion`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is

configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## fork

This event occurs when someone forks a repository. For more information, see "Fork a repo." For information about the API to manage forks, see "Forks" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Contents" repository permission.

### Availability for `fork`

- Enterprises
- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `fork`

#### Webhook request body parameters

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`forkee`** object Required

The created `repository` resource.

▶ Properties of `forkee`

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**sender** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## github_app_authorization

This event occurs when a user revokes their authorization of a GitHub App. For more information, see "About apps." For information about the API to manage GitHub Apps, see the GraphQL API documentation or "Apps" in the REST API documentation.

A GitHub App receives this webhook by default and cannot unsubscribe from this event.

Anyone can revoke their authorization of a GitHub App from their GitHub account settings page. Revoking the authorization of a GitHub App does not uninstall the GitHub App. You should program your GitHub App so that when it receives this webhook, it stops calling the API on behalf of the person who revoked the token. If your GitHub App continues to use a revoked access token, it will receive the `401 Bad Credentials` error. For details about requests with a user access token, which require GitHub App authorization, see "Authenticating with a GitHub App on behalf of a user."

### Availability for `github_app_authorization`

- GitHub Apps

### Webhook payload object for `github_app_authorization`

Someone revoked their authorization of a GitHub App.

#### Webhook request body parameters

**action** string Required

Value: `revoked`

**enterprise** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**installation** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**organization** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**repository** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**sender** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# gollum

This event occurs when someone creates or updates a wiki page. For more information, see "About wikis."

To subscribe to this event, a GitHub App must have at least read-level access for the "Contents" repository permission.

## Availability for `gollum`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `gollum`

#### Webhook request body parameters

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`pages`** array of objects Required

The pages that were updated.

▶ Properties of `pages`

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# installation

This event occurs when there is activity relating to a GitHub App installation. All GitHub Apps receive this event by default. You cannot manually subscribe to this event.

For more information about GitHub Apps, see "About apps." For information about the APIs to manage GitHub Apps, see the GraphQL API documentation or "Apps" in the REST API documentation.

### Availability for `installation`

- GitHub Apps

### Webhook payload object for `installation`

Action type: **created**

Someone installed a GitHub App on a user or organization account.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object Required

Installation

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repositories`** array of objects

An array of repository objects that the installation can access.

▶ Properties of `repositories`

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`requester`** object or null

▶ Properties of `requester`

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## installation_repositories

This event occurs when there is activity relating to which repositories a GitHub App installation can access. All GitHub Apps receive this event by default. You cannot manually subscribe to this event.

For more information about GitHub Apps, see "About apps." For information about the APIs to manage GitHub Apps, see the GraphQL API documentation or "Apps" in the REST API documentation.

## Availability for `installation_repositories`

- GitHub Apps

## Webhook payload object for `installation_repositories`

Action type: **added**

A GitHub App installation was granted access to one or more repositories.

#### Webhook request body parameters

**`action`** string Required

Value: `added`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object Required

Installation

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repositories_added`** array of objects Required

An array of repository objects, which were added to the installation.

▶ Properties of `repositories_added`

**`repositories_removed`** array of objects Required

An array of repository objects, which were removed from the installation.

▶ Properties of `repositories_removed`

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`repository_selection`** string Required

Describe whether all repositories have been selected or there's a selection involved

Can be one of: `all`, `selected`

**`requester`** object or null Required

▶ Properties of `requester`

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# installation_target

This event occurs when there is activity relating to the user or organization account that a GitHub App is installed on. For more information, see "About apps." For information about the APIs to manage GitHub Apps, see the GraphQL API documentation or "Apps" in the REST API documentation.

## Availability for `installation_target`

- GitHub Apps

## Webhook payload object for `installation_target`

Somebody renamed the user or organization account that a GitHub App is installed on.

#### Webhook request body parameters

**`account`** object Required

▶ Properties of `account`

**`action`** string Required

Value: `renamed`

**`changes`** object Required

▶ Properties of `changes`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object Required

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

**`target_type`** string Required

# issue_comment

This event occurs when there is activity relating to a comment on an issue or pull request. For more information about issues and pull requests, see "About issues" and "About pull requests." For information about the APIs to manage issue comments, see the GraphQL documentation or "Issue comments" in the REST API documentation.

For activity relating to an issue as opposed to comments on an issue, use the `issue` event. For activity related to pull request reviews or pull request review comments, use the `pull_request_review` or `pull_request_review_comment` events. For more information about the different types of pull request comments, see "Working with comments."

To subscribe to this event, a GitHub App must have at least read-level access for the "Issues" repository permission.

## Availability for `issue_comment`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `issue_comment`

Action type: **created**

A comment on an issue or pull request was created.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`comment`** object Required

The comment itself.

▶ Properties of `comment`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`issue`** object Required

The issue the comment belongs to.

▶ Properties of `issue`

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is

configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## issues

This event occurs when there is activity relating to an issue. For more information about issues, see "About issues." For information about the APIs to manage issues, see the GraphQL documentation or "Issues" in the REST API documentation.

For activity relating to a comment on an issue, use the `issue_comment` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Issues" repository permission.

### Availability for `issues`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `issues`

Action type: **assigned**

An issue was assigned to a user.

#### Webhook request body parameters

**`action`** string Required

The action that was performed.

Value: `assigned`

**`assignee`** object or null

▶ Properties of `assignee`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`issue`** object Required

The issue itself.

▶ Properties of `issue`

---

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## label

This event occurs when there is activity relating to labels. For more information, see "Managing labels." For information about the APIs to manage labels, see the GraphQL documentation or "Labels" in the REST API documentation.

If you want to receive an event when a label is added to or removed from an issue, pull request, or discussion, use the `labeled` or `unlabeled` action type for the `issues`, `pull_request`, or `discussion` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Metadata" repository permission.

### Availability for `label`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `label`

Action type: **created**

A label was created.

#### Webhook request body parameters

---

**`action`** string Required

Value: `created`

---

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

---

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

`label` object Required

▶ Properties of `label`

`organization` object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

`repository` object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

`sender` object

The GitHub user that triggered the event. This property is included in every webhook payload.

## marketplace_purchase

This event occurs when there is activity relating to a GitHub Marketplace purchase. For more information, see "GitHub Marketplace." For information about the APIs to manage GitHub Marketplace listings, see the GraphQL documentation or "GitHub Marketplace" in the REST API documentation.

### Availability for `marketplace_purchase`

- GitHub Marketplace

### Webhook payload object for `marketplace_purchase`

Action type: **cancelled**

Someone cancelled a GitHub Marketplace plan, and the last billing cycle has ended. The change will take effect on the account immediately.

#### Webhook request body parameters

`action` string Required

Value: `cancelled`

`effective_date` string Required

`enterprise` object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

`installation` object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

`marketplace_purchase` object Required

▶ Properties of `marketplace_purchase`

---

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`previous_marketplace_purchase`** object

▶ Properties of `previous_marketplace_purchase`

---

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## member

This event occurs when there is activity relating to collaborators in a repository. For more information, see "Adding outside collaborators to repositories in your organization." For more information about the API to manage repository collaborators, see the GraphQL API documentation or "Collaborators" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Members" organization permission.

### Availability for `member`

- Enterprises
- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `member`

Action type: **added**

A GitHub user accepted an invitation to a repository.

#### Webhook request body parameters

---

**`action`** string Required

Value: `added`

---

**`changes`** object

▶ Properties of `changes`

---

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For

configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`member`** object or null Required

▶ Properties of `member`

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## membership

This event occurs when there is activity relating to team membership. For more information, see "About teams." For more information about the APIs to manage team memberships, see the GraphQL API documentation or "Team members" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Members" organization permission.

### Availability for `membership`

- Organizations
- Enterprises
- GitHub Apps

### Webhook payload object for `membership`

Action type: **added**

An organization member was added to a team.

#### Webhook request body parameters

**`action`** string Required

Value: `added`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For

configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`member`** object or null Required

▶ Properties of `member`

**`organization`** object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`scope`** string Required

The scope of the membership. Currently, can only be `team`.

Value: `team`

**`sender`** object or null Required

**`team`** object Required

Groups of organization members that gives permissions on specified repositories.

▶ Properties of `team`

## merge_group

This event occurs when there is activity relating to a merge group in a merge queue. For more information, see "Managing a merge queue."

To subscribe to this event, a GitHub App must have at least read-level access for the "Merge queues" repository permission.

### Availability for `merge_group`

- GitHub Apps

### Webhook payload object for `merge_group`

Action type: **checks_requested**

Status checks were requested for a merge group. This happens when a merge group is created or added to by the merge queue because a pull request was queued.

When you receive this event, you should perform checks on the head SHA and report status back using check runs or commit statuses.

**Webhook request body parameters**

**`action`** string Required

Value: `checks_requested`

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`merge_group`** object Required

A group of pull requests that the merge queue has grouped together to be merged.

▶ Properties of `merge_group`

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

# meta

This event occurs when there is activity relating to a webhook itself.

To subscribe to this event, a GitHub App must have at least read-level access for the "Meta" app permission.

## Availability for `meta`

- GitHub Marketplace
- Enterprises
- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `meta`

The webhook was deleted.

**Webhook request body parameters**

**`action`** string Required

Value: `deleted`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is

configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`hook`** object Required

The modified webhook. This will contain different keys based on the type of webhook it is: repository, organization, business, app, or GitHub Marketplace.

▶ Properties of `hook`

**`hook_id`** integer Required

The id of the modified webhook.

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## milestone

This event occurs when there is activity relating to milestones. For more information, see "About milestones." For information about the APIs to manage milestones, see the GraphQL documentation or "Milestones" in the REST API documentation.

If you want to receive an event when an issue or pull request is added to or removed from a milestone, use the `milestoned` or `demilestoned` action type for the `issues` or `pull_request` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Issues" or "Pull requests" repository permissions.

### Availability for `milestone`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `milestone`

Action type: **closed**

A milestone was closed.

**Webhook request body parameters**

**`action`** string Required

Value: `closed`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`milestone`** object Required

A collection of related issues and pull requests.

▶ Properties of `milestone`

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# org_block

This event occurs when organization owners or moderators block or unblock a non-member from collaborating on the organization's repositories. For more information, see "Blocking a user from your organization." For information about the APIs to manage blocked users, see the GraphQL documentation or "Blocking users" in the REST API documentation.

If you want to receive an event when members are added or removed from an organization, use the `organization` event instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Administration" organization permission.

## Availability for `org_block`

- Organizations
- Enterprises
- GitHub Apps

## Webhook payload object for `org_block`

Action type: **blocked**

Action type: **blocked**

A user was blocked from the organization.

**Webhook request body parameters**

`action` string Required

Value: `blocked`

`blocked_user` object or null Required

▶ Properties of `blocked_user`

`enterprise` object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

`installation` object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

`organization` object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

`repository` object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

`sender` object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## organization

This event occurs when there is activity relating to an organization and its members. For more information, see "About organizations." For information about the APIs to manage organizations, see the GraphQL documentation or "Organizations" in the REST API documentation.

If you want to receive an event when a non-member is blocked or unblocked from an organization, use the `org_block` event instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Members" organization permission.

### Availability for `organization`

- Organizations
- Enterprises
- GitHub Apps

### Webhook payload object for `organization`

Action type: **deleted**

An organization was deleted.

#### Webhook request body parameters

**`action`** string Required

Value: `deleted`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`membership`** object

The membership between the user and the organization. Not present when the action is `member_invited`.

▶ Properties of `membership`

**`organization`** object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## package

This event occurs when there is activity relating to GitHub Packages. For more information, see "Introduction to GitHub Packages." For information about the APIs to manage GitHub Packages, see the GraphQL API documentation or "Packages" in the REST API documentation.

To install this event on a GitHub App, the app must have at least read-level access for the "Packages" repository permission.

### Availability for `package`

- Repositories
- Organizations

- GitHub Apps

### Webhook payload object for `package`

Action type: **published**

A package was published to a registry.

#### Webhook request body parameters

**`action`** string Required

Value: `published`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`package`** object Required

Information about the package.

▶ Properties of `package`

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## page_build

This event occurs when there is an attempted build of a GitHub Pages site. This event occurs regardless of whether the build is successful. For more information, see "Configuring a publishing source for your GitHub Pages site." For information about the API to manage GitHub Pages, see "Pages" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Pages" repository permission.

### Availability for `page_build`

- Repositories

- Organizations
- GitHub Apps

### Webhook payload object for `page_build`

#### Webhook request body parameters

**`build`** object Required

The List GitHub Pages builds itself.

▶ Properties of `build`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`id`** integer Required

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## personal_access_token_request

This event occurs when there is activity relating to a request for a fine-grained personal access token to access resources that belong to a resource owner that requires approval for token access. For more information, see "Creating a personal access token."

To subscribe to this event, a GitHub App must have at least read-level access for the "Personal access token requests" organization permission.

**Note**: Fine-grained PATs are in public beta. Related APIs, events, and functionality are subject to change.

### Availability for `personal_access_token_request`

- GitHub Apps

### Webhook payload object for `personal_access_token_request`

Action type: **approved**

A fine-grained personal access token request was approved.

#### Webhook request body parameters

**`action`** string Required

Value: `approved`

**`personal_access_token_request`** object Required

Details of a Personal Access Token Request.

▶ Properties of `personal_access_token_request`

**`organization`** object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`installation`** object Required

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

## ping

This event occurs when you create a new webhook. The ping event is a confirmation from GitHub that you configured the webhook correctly.

### Availability for `ping`

- Repositories
- Organizations
- GitHub Apps
- Enterprises
- GitHub Marketplace

### Webhook payload object for `ping`

#### Webhook request body parameters

**`hook`** object

The webhook that is being pinged

▶ Properties of `hook`

**`hook_id`** integer

The ID of the webhook that triggered the ping.

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

**`zen`** string

Random string of GitHub zen.

## project_card

This event occurs when there is activity relating to a card on a classic project. For more information, see "About projects (classic)." For information about the API to manage classic projects, see the GraphQL API documentation or "Projects (classic)" in the REST API documentation.

For activity relating to a project or a column on a project, use the `project` and `project_column` event. For activity relating to Projects instead of Projects (classic), use the `projects_v2` event instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Projects" repository or organization permission.

### Availability for `project_card`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `project_card`

Action type: **converted**

A note in a classic project was converted to an issue.

#### Webhook request body parameters

**`action`** string Required

Value: `converted`

**`changes`** object Required

▶ Properties of `changes`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is

configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`project_card`** object Required

▶ Properties of `project_card`

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## project

This event occurs when there is activity relating to a classic project. For more information, see "About projects (classic)." For information about the API to manage classic projects, see the GraphQL API documentation or "Projects (classic)" in the REST API documentation.

For activity relating to a card or column on a project, use the `project_card` and `project_column` event. For activity relating to Projects instead of Projects (classic), use the `projects_v2` event instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Projects" repository or organization permission.

### Availability for `project`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `project`

Action type: **closed**

A classic project was closed.

#### Webhook request body parameters

**`action`** string Required

Value: `closed`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`project`** object Required

▶ Properties of `project`

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## project_column

This event occurs when there is activity relating to a column on a classic project. For more information, see "About projects (classic)." For information about the API to manage classic projects, see the GraphQL API documentation or "Projects (classic)" in the REST API documentation.

For activity relating to a project or a card on a project, use the `project` and `project_card` event. For activity relating to Projects instead of Projects (classic), use the `projects_v2` event instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Projects" repository or organization permission.

### Availability for `project_column`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `project_column`

Action type: **created**

A column was added to a classic project.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

---

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

---

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

---

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`project_column`** object Required

▶ Properties of `project_column`

---

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## projects_v2

This event occurs when there is activity relating to an organization-level project. For more information, see "About Projects." For information about the Projects API, see the GraphQL documentation.

For activity relating to a item on a project, use the `projects_v2_item` event. For activity relating to Projects (classic), use the `project`, `project_card`, and `project_column` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Projects" organization permission.

**Note**: Webhook events for projects are currently in beta and subject to change. To share feedback about projects webhooks with GitHub, see the Projects webhook feedback discussion.

### Availability for `projects_v2`

- Organizations

### Webhook payload object for `projects_v2`

Action type: **closed**

A project in the organization was closed.

**Webhook request body parameters**

**`action`** string Required

Value: `closed`

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`projects_v2`** object Required

A projects v2 project

▶ Properties of `projects_v2`

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## projects_v2_item

This event occurs when there is activity relating to an item on an organization-level project. For more information, see "About Projects." For information about the Projects API, see the GraphQL documentation.

For activity relating to a project (instead of an item on a project), use the `projects_v2` event. For activity relating to Projects (classic), use the `project`, `project_card`, and `project_column` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Projects" organization permission.

**Note**: Webhook events for projects are currently in beta and subject to change. To share feedback about projects webhooks with GitHub, see the Projects webhook feedback discussion.

### Availability for `projects_v2_item`

- Organizations

### Webhook payload object for `projects_v2_item`

Action type: **archived**

An item on an organization project was archived. For more information, see "Archiving items from your project."

**Webhook request body parameters**

**`action`** string Required

Value: `archived`

value: `archived`

**`changes`** object Required

▶ Properties of `changes`

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`projects_v2_item`** object Required

An item belonging to a project

▶ Properties of `projects_v2_item`

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## public

This event occurs when repository visibility changes from private to public. For more information, see "Setting repository visibility."

To subscribe to this event, a GitHub App must have at least read-level access for the "Metadata" repository permission.

### Availability for `public`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `public`

#### Webhook request body parameters

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is

configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## pull_request

This event occurs when there is activity on a pull request. For more information, see "About pull requests." For information about the APIs to manage pull requests, see the GraphQL API documentation or "Pulls" in the REST API documentation.

For activity related to pull request reviews, pull request review comments, pull request comments, or pull request review threads, use the `pull_request_review`, `pull_request_review_comment`, `issue_comment`, or `pull_request_review_thread` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Pull requests" repository permission.

### Availability for `pull_request`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `pull_request`

Action type: **assigned**

A pull request was assigned to a user.

#### Webhook request body parameters

**`action`** string Required

Value: `assigned`

**`assignee`** object or null Required

▶ Properties of `assignee`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

Apps.

**number** integer Required

The pull request number.

**organization** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**pull_request** object Required

▶ Properties of `pull_request`

**repository** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**sender** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## pull_request_review_comment

This event occurs when there is activity relating to a pull request review comment. A pull request review comment is a comment on a pull request's diff. For more information, see "Commenting on a pull request." For information about the APIs to manage pull request review comments, see the GraphQL API documentation or "Pull request review comments" in the REST API documentation.

For activity related to pull request reviews, pull request comments, or pull request review threads, use the `pull_request_review`, `issue_comment`, or `pull_request_review_thread` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Pull requests" repository permission.

### Availability for `pull_request_review_comment`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `pull_request_review_comment`

Action type: **created**

A comment on a pull request diff was created.

#### Webhook request body parameters

**action** string Required

Value: `created`

**comment** object Required

The comment itself

The comment itself.

▶ Properties of `comment`

---

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

---

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

---

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`pull_request`** object Required

▶ Properties of `pull_request`

---

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## pull_request_review

This event occurs when there is activity relating to a pull request review. A pull request review is a group of pull request review comments in addition to a body comment and a state. For more information, see "About pull request reviews." For information about the APIs to manage pull request reviews, see the GraphQL API documentation or "Pull request reviews" in the REST API documentation.

For activity related to pull request review comments, pull request comments, or pull request review threads, use the `pull_request_review_comment`, `issue_comment`, or `pull_request_review_thread` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Pull requests" repository permission.

### Availability for `pull_request_review`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `pull_request_review`

Action type: **dismissed**

A review on a pull request was dismissed.

#### Webhook request body parameters

**`action`** string Required

Value: `dismissed`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`pull_request`** object Required

▶ Properties of `pull_request`

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`review`** object Required

The review that was affected.

▶ Properties of `review`

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## pull_request_review_thread

This event occurs when there is activity relating to a comment thread on a pull request. For more information, see "About pull request reviews." For information about the APIs to manage pull request reviews, see the GraphQL API documentation or "Pull request review comments" in the REST API documentation.

For activity related to pull request review comments, pull request comments, or pull request reviews, use the `pull_request_review_comment`, `issue_comment`, or `pull_request_review` events instead.

To subscribe to this event, a GitHub App must have at least read-level access for the "Pull requests" repository permission.

### Availability for `pull_request_review_thread`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `pull_request_review_thread`

Action type: **resolved**

A comment thread on a pull request was marked as resolved.

#### Webhook request body parameters

**`action`** string Required

Value: `resolved`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`pull_request`** object Required

▶ Properties of `pull_request`

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

**`thread`** object Required

▶ Properties of `thread`

## push

This event occurs when there is a push to a repository branch. This includes when a commit is pushed, when a commit tag is pushed, when a branch is deleted, when a tag is deleted, or when a repository is created from a template. To subscribe to only branch and tag deletions, use the `delete` webhook event.

To subscribe to this event, a GitHub App must have at least read-level access for the

"Contents" repository permission.

**Note**: An event will not be created when more than three tags are pushed at once.

## Availability for `push`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `push`

#### Webhook request body parameters

**`after`** string Required

The SHA of the most recent commit on `ref` after the push.

**`base_ref`** string or null Required

**`before`** string Required

The SHA of the most recent commit on `ref` before the push.

**`commits`** array of objects Required

An array of commit objects describing the pushed commits. (Pushed commits are all commits that are included in the `compare` between the `before` commit and the `after` commit.) The array includes a maximum of 20 commits. If necessary, you can use the Commits API to fetch additional commits. This limit is applied to timeline events only and isn't applied to webhook deliveries.

▶ Properties of `commits`

**`compare`** string Required

URL that shows the changes in this `ref` update, from the `before` commit to the `after` commit. For a newly created `ref` that is directly based on the default branch, this is the comparison between the head of the default branch and the `after` commit. Otherwise, this shows all commits until the `after` commit.

**`created`** boolean Required

Whether this push created the `ref`.

**`deleted`** boolean Required

Whether this push deleted the `ref`.

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`forced`** boolean Required

Whether this push was a force push of the `ref`.

**`head_commit`** object or null Required

▶ Properties of `head_commit`

▶ Properties of `head_commit`

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`pusher`** object Required

Metaproperties for Git author/committer information.

▶ Properties of `pusher`

**`ref`** string Required

The full git ref that was pushed. Example: `refs/heads/main` or `refs/tags/v3.14.1`.

**`repository`** object Required

A git repository

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## registry_package

This event occurs when there is activity relating to GitHub Packages. For more information, see "Introduction to GitHub Packages." For information about the APIs to manage GitHub Packages, see the GraphQL API documentation or "Packages" in the REST API documentation.

To install this event on a GitHub App, the app must have at least read-level access for the "Packages" repository permission.

**Note**: GitHub recommends that you use the newer `package` event instead.

### Availability for `registry_package`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `registry_package`

Action type: **published**

A package was published to a registry.

#### Webhook request body parameters

**`action`** string Required

Value: `published`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`registry_package`** object Required

▶ Properties of `registry_package`

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# release

This event occurs when there is activity relating to releases. For more information, see "About releases." For information about the APIs to manage releases, see the GraphQL API documentation or "Releases" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Contents" repository permission.

## Availability for `release`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `release`

Action type: **created**

A draft was saved, or a release or pre-release was published without previously being saved as a draft.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`release`** object Required

The release object.

▶ Properties of `release`

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## repository_advisory

This event occurs when there is activity relating to a repository security advisory. For more information about repository security advisories, see "About GitHub Security Advisories for repositories."

To subscribe to this event, a GitHub App must have at least read-level access for the "Repository security advisories" permission.

### Availability for `repository_advisory`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `repository_advisory`

Action type: **published**

A repository security advisory was published.

#### Webhook request body parameters

**`action`** string Required

Value: `published`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`repository_advisory`** object Required

A repository security advisory.

▶ Properties of `repository_advisory`

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## repository

This event occurs when there is activity relating to repositories. For more information, see "About repositories." For information about the APIs to manage repositories, see the GraphQL documentation or "Repositories" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Metadata" repository permission.

### Availability for `repository`

- Enterprises
- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `repository`

Action type: **archived**

A repository was archived.

#### Webhook request body parameters

**`action`** string Required

Value: `archived`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## repository_dispatch

This event occurs when a GitHub App sends a `POST` request to `/repos/{owner}/{repo}/dispatches`. For more information, see the REST API documentation for creating a repository dispatch event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Contents" repository permission.

### Availability for `repository_dispatch`

- GitHub Apps

### Webhook payload object for `repository_dispatch`

The `event_type` that was specified in the `POST /repos/{owner}/{repo}/dispatches` request body.

#### Webhook request body parameters

**`action`** string Required

Value: `sample.collected`

**`branch`** string Required

**`client_payload`** object or null Required

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object Required

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## repository_import

This event occurs when a repository is imported to GitHub Enterprise Cloud. For more information, see "Importing a repository with GitHub Importer." For more information about the API to manage imports, see the REST API documentation.

### Availability for `repository_import`

- Repositories
- Organizations

### Webhook payload object for `repository_import`

#### Webhook request body parameters

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`status`** string Required

Can be one of: `success`, `cancelled`, `failure`

## repository_ruleset

This event occurs when there is activity relating to repository rulesets. For more information about repository rulesets, see "Managing rulesets." For more information on managing rulesets via the APIs, see Repository ruleset in the GraphQL documentation or "Repository rules" and "Organization rules in the REST API documentation."

To subscribe to this event, a GitHub App must have at least read-level access for the "Administration" repository or organization permission.

### Availability for `repository_ruleset`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `repository_ruleset`

Action type: **created**

A repository ruleset was created.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`repository_ruleset`** object Required

A set of rules to apply when specified conditions are met.

▶ Properties of `repository_ruleset`

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## repository_vulnerability_alert

This event occurs when there is activity relating to a security vulnerability alert in a repository.

**Note**: This event is deprecated. Use the `dependabot_alert` event instead.

### Availability for `repository_vulnerability_alert`

- Repositories
- Organizations

### Webhook payload object for `repository_vulnerability_alert`

Action type: **create**

A repository vulnerability alert was created.

#### Webhook request body parameters

**`action`** string Required

Value: `create`

**`alert`** object Required

▶ Properties of `alert`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

# secret_scanning_alert

This event occurs when there is activity relating to a secret scanning alert. For more information about secret scanning, see "About secret scanning." For information about the API to manage secret scanning alerts, see "Secret scanning" in the REST API documentation.

For activity relating to secret scanning alert locations, use the `secret_scanning_alert_location` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Secret scanning alerts" repository permission.

## Availability for `secret_scanning_alert`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `secret_scanning_alert`

Action type: **created**

A secret scanning alert was created.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`alert`** object Required

▶ Properties of `alert`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object

**sender** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## secret_scanning_alert_location

This event occurs when there is activity relating to the locations of a secret in a secret scanning alert.

For more information about secret scanning, see "About secret scanning." For information about the API to manage secret scanning alerts, see "Secret scanning" in the REST API documentation.

For activity relating to secret scanning alerts, use the `secret_scanning_alert` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Secret scanning alerts" repository permission.

### Availability for `secret_scanning_alert_location`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `secret_scanning_alert_location`

A new instance of a previously detected secret was detected in a repository, and the location of the secret was added to the existing alert.

#### Webhook request body parameters

**`action`** string

Value: `created`

**`alert`** object Required

▶ Properties of `alert`

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`location`** object Required

▶ Properties of `location`

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**sender** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## security_advisory

This event occurs when there is activity relating to a security advisory that was reviewed by GitHub. A GitHub-reviewed security advisory provides information about security-related vulnerabilities in software on GitHub. For more information about security advisories, see "About GitHub Security Advisories for repositories." For information about the API to manage security advisories, see the GraphQL documentation.

GitHub Dependabot alerts are also powered by the security advisory dataset. For more information, see "About Dependabot alerts."

### Availability for `security_advisory`

- GitHub Apps

### Webhook payload object for `security_advisory`

Action type: **published**

A security advisory was published to the GitHub community.

#### Webhook request body parameters

**action** string Required

Value: `published`

**enterprise** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**installation** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**organization** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**repository** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**security_advisory** object Required

The details of the security advisory, including summary, description, and severity.

▶ Properties of `security_advisory`

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## security_and_analysis

This event occurs when code security and analysis features are enabled or disabled for a repository. For more information, see "GitHub security features."

To subscribe to this event, a GitHub App must have at least read-level access for the "Administration" repository permission.

### Availability for `security_and_analysis`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `security_and_analysis`

#### Webhook request body parameters

**`changes`** object Required

▶ Properties of `changes`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

Full Repository

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

## sponsorship

This event occurs when there is activity relating to a sponsorship listing. For more information, see "About GitHub Sponsors." For information about the API to manage sponsors, see the GraphQL documentation.

You can only create a sponsorship webhook on GitHub.com. For more information, see

"Configuring webhooks for events in your sponsored account."

## Availability for `sponsorship`

- Sponsored accounts

## Webhook payload object for `sponsorship`

Action type: **cancelled**

A sponsorship was cancelled and the last billing cycle has ended.

This event is only sent when a recurring (monthly) sponsorship is cancelled; it is not sent for one-time sponsorships.

#### Webhook request body parameters

**`action`** string Required

Value: `cancelled`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`sponsorship`** object Required

▶ Properties of `sponsorship`

# star

This event occurs when there is activity relating to repository stars. For more information about stars, see "Saving repositories with stars." For information about the APIs to manage stars, see the GraphQL documentation or "Starring" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the

"Metadata" repository permission.

### Availability for `star`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `star`

Action type: **created**

Someone starred a repository.

#### Webhook request body parameters

**`action`** string Required

Value: `created`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`starred_at`** string or null Required

The time the star was created. This is a timestamp in ISO 8601 format: `YYYY-MM-DDTHH:MM:SSZ`. Will be `null` for the `deleted` action.

## status

This event occurs when the status of a Git commit changes. For example, commits can be marked as `error`, `failure`, `pending`, or `success`. For more information, see "About status checks." For information about the APIs to manage commit statuses, see the GraphQL documentation or "Commit statuses" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the

"Commit statuses" repository permission.

## Availability for `status`

- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `status`

#### Webhook request body parameters

**`avatar_url`** string or null

**`branches`** array of objects Required

An array of branch objects containing the status' SHA. Each branch contains the given SHA, but the SHA may or may not be the head of the branch. The array includes a maximum of 10 branches.

▶ Properties of `branches`

**`commit`** object Required

▶ Properties of `commit`

**`context`** string Required

**`created_at`** string Required

**`description`** string or null Required

The optional human-readable description added to the status.

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`id`** integer Required

The unique identifier of the status.

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`name`** string Required

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`sha`** string Required

The Commit SHA.

**`state`** string Required

The new state. Can be `pending`, `success`, `failure`, or `error`.

Can be one of: `pending`, `success`, `failure`, `error`

**`target_url`** string or null Required

The optional link added to the status.

**`updated_at`** string Required

## team_add

This event occurs when a team is added to a repository. For more information, see "Managing teams and people with access to your repository."

For activity relating to teams, see the `teams` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Members" organization permission.

### Availability for `team_add`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `team_add`

#### Webhook request body parameters

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository`

property when the event occurs from activity in a repository.

**sender** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**team** object Required

Groups of organization members that gives permissions on specified repositories.

▶ Properties of `team`

## team

This event occurs when there is activity relating to teams in an organization. For more information, see "About teams."

To subscribe to this event, a GitHub App must have at least read-level access for the "Members" organization permission.

### Availability for `team`

- Organizations
- Enterprises
- GitHub Apps

### Webhook payload object for `team`

Action type: **added_to_repository**

A team was granted access to a repository.

#### Webhook request body parameters

**action** string Required

Value: `added_to_repository`

**enterprise** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**installation** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**organization** object Required

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**repository** object

A git repository

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

---

**`team`** object Required

Groups of organization members that gives permissions on specified repositories.

▶ Properties of `team`

## user

This event occurs when there is activity relating to user accounts in an enterprise.

### Availability for `user`

- Enterprises

### Webhook payload object for `user`

Action type: **created**

A user account was added to the enterprise.

#### Webhook request body parameters

---

**`action`** string Required

Value: `created`

---

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

---

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

---

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

---

**`repository`** object

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

---

**`sender`** object

The GitHub user that triggered the event. This property is included in every webhook payload.

---

**`user`** object or null

▶ Properties of `user`

## watch

This event occurs when there is activity relating to watching, or subscribing to, a repository. For more information about watching, see "Managing your subscriptions." For information about the APIs to manage watching, see "Watching" in the REST API documentation.

To subscribe to this event, a GitHub App must have at least read-level access for the "Metadata" repository permission.

### Availability for `watch`

- Repositories
- Organizations
- GitHub Apps

### Webhook payload object for `watch`

Someone started watching the repository.

#### Webhook request body parameters

**`action`** string Required

Value: `started`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

## workflow_dispatch

This event occurs when a GitHub Actions workflow is manually triggered. For more information, see "Manually running a workflow."

For activity relating to workflow runs, use the `workflow_run` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Contents" repository permission.

### Availability for `workflow_dispatch`

- GitHub Apps

### Webhook payload object for `workflow_dispatch`

#### Webhook request body parameters

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`inputs`** object or null Required

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`ref`** string Required

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`workflow`** string Required

## workflow_job

This event occurs when there is activity relating to a job in a GitHub Actions workflow. For more information, see "Using jobs in a workflow." For information about the API to manage workflow jobs, see "Workflow jobs" in the REST API documentation.

For activity relating to a workflow run instead of a job in a workflow run, use the `workflow_run` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Actions" repository permission.

## Availability for `workflow_job`

- Enterprises
- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `workflow_job`

Action type: **completed**

A job in a workflow run finished. This event occurs when a job in a workflow is completed, regardless of whether the job was successful or unsuccessful.

#### Webhook request body parameters

**`action`** string Required

Value: `completed`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`workflow_job`** object Required

▶ Properties of `workflow_job`

**`deployment`** object

A request for a specific ref(branch,sha,tag) to be deployed

▶ Properties of `deployment`

## workflow_run

This event occurs when there is activity relating to a run of a GitHub Actions workflow. For more information, see "About workflows." For information about the APIs to manage workflow runs, see the GraphQL documentation or "Workflow runs" in the REST API documentation.

For activity relating to a job in a workflow run, use the `workflow_job` event.

To subscribe to this event, a GitHub App must have at least read-level access for the "Actions" repository permission.

## Availability for `workflow_run`

- Enterprises
- Repositories
- Organizations
- GitHub Apps

## Webhook payload object for `workflow_run`

Action type: **completed**

A workflow run finished. This event occurs when a workflow run is completed, regardless of whether the workflow was successful or unsuccessful.

### Webhook request body parameters

**`action`** string Required

Value: `completed`

**`enterprise`** object

An enterprise on GitHub. Webhook payloads contain the `enterprise` property when the webhook is configured on an enterprise account or an organization that's part of an enterprise account. For more information, see "About enterprise accounts."

**`installation`** object

The GitHub App installation. Webhook payloads contain the `installation` property when the event is configured for and sent to a GitHub App. For more information, see "Using webhooks with GitHub Apps."

**`organization`** object

A GitHub organization. Webhook payloads contain the `organization` property when the webhook is configured for an organization, or when the event occurs from activity in a repository owned by an organization.

**`repository`** object Required

The repository on GitHub where the event occurred. Webhook payloads contain the `repository` property when the event occurs from activity in a repository.

**`sender`** object Required

The GitHub user that triggered the event. This property is included in every webhook payload.

**`workflow`** object or null Required

▶ Properties of `workflow`

`workflow_run` object Required

▶ Properties of `workflow_run`

**Legal**

© 2023 GitHub, Inc. Terms Privacy Status Pricing Expert services Blog