

Handling failed webhook deliveries

GitHub does not automatically redeliver failed webhook deliveries, but you can handle failed deliveries manually or by writing code.

In this article

About webhook delivery failures

Handling delivery failures

About webhook delivery failures [↗](#)

A webhook delivery can fail for multiple reasons. For example, if your server is down or takes longer than 30 seconds to respond, GitHub will record the delivery as a failure.

GitHub does not automatically redeliver failed deliveries.

Handling delivery failures [↗](#)

You can manually redeliver failed deliveries. For more information, see "[Redelivering webhooks](#)."

You can also write a script that checks for failed deliveries and attempts to redeliver any that failed. Your script should run on a schedule and do the following:

- 1 Use the GitHub REST API to fetch data about any webhook deliveries that were attempted since the last time that your script ran. For more information, see "[Repository webhooks](#)," "[Organization webhooks](#)," and "[GitHub App webhooks](#)."

There are no API endpoints to get data about global webhook deliveries.

- 2 Look at the fetched data to see if any deliveries failed. The data for a failed delivery will have a `status` value that is not `OK`.
- 3 Use the GitHub REST API to redeliver any deliveries that failed. For more information, see "[Repository webhooks](#)," "[Organization webhooks](#)," and "[GitHub App webhooks](#)."

For example scripts, see:

- "[Automatically redelivering failed deliveries for a repository webhook](#)"
- "[Automatically redelivering failed deliveries for an organization webhook](#)"
- "[Automatically redelivering failed deliveries for a GitHub App webhook](#)"

If a webhook delivery fails repeatedly, you should investigate the cause. Each failed delivery will give a reason for failure. For more information, see "[Troubleshooting webhooks](#)."

Legal