

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

Check Runs

Use the REST API to manage check runs.

Create a check run

✔ Works with [GitHub Apps](#)

Note: The Checks API only looks for pushes in the repository where the check suite or check run were created. Pushes to a branch in a forked repository are not detected and return an empty `pull_requests` array.

Creates a new check run for a specific commit in a repository. Your GitHub App must have the `checks:write` permission to create check runs.

In a check suite, GitHub limits the number of check runs with the same name to 1000. Once these check runs exceed 1000, GitHub will start to automatically delete older check runs.

Parameters for "Create a check run"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string **Required**

The account owner of the repository. The name is not case sensitive.

`repo` string **Required**

The name of the repository without the `.git` extension. The name is not case sensitive.

Body parameters

`name` string **Required**

The name of the check. For example, "code-coverage".

`head_sha` string **Required**

The SHA of the commit.

`details_url` string

The URL of the integrator's site that has the full details of the check. If the integrator does not provide this, then the homepage of the GitHub app is used.

`external_id` string

A reference for the run on the integrator's system.

status string

The current status.

Default: `queued`

Can be one of: `queued`, `in_progress`, `completed`

started_at string

The time that the check run began. This is a timestamp in [ISO 8601](#) format: `YYYY-MM-DDTHH:MM:SSZ`.

conclusion string

Required if you provide `completed_at` or a `status` of `completed`. The final conclusion of the check. **Note:** Providing `conclusion` will automatically set the `status` parameter to `completed`. You cannot change a check run conclusion to `stale`, only GitHub can set this.

Can be one of: `action_required`, `cancelled`, `failure`, `neutral`, `success`, `skipped`, `stale`, `timed_out`

completed_at string

The time the check completed. This is a timestamp in [ISO 8601](#) format: `YYYY-MM-DDTHH:MM:SSZ`.

output object

Check runs can accept a variety of data in the `output` object, including a `title` and `summary` and can optionally provide descriptive details about the run.

► Properties of `output`

actions array of objects

Displays a button on GitHub that can be clicked to alert your app to do additional tasks. For example, a code linting app can display a button that automatically fixes detected errors. The button created in this object is displayed after the check run completes. When a user clicks the button, GitHub sends the [check_run.requested_action webhook](#) to your app. Each action includes a `label`, `identifier` and `description`. A maximum of three actions are accepted. To learn more about check runs and requested actions, see "[Check runs and requested actions](#)."

► Properties of `actions`

HTTP response status codes for "Create a check run"

Status code	Description
201	Created

Code samples for "Create a check run"

Example of an `in_progress` conclusion ▾

POST

/repos/{owner}/{repo}/check-runs

cURL

JavaScript

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/check-runs \ -d '{ "name": "mighty_readme", "head_sha": "ce587453ced02b1526dfb4cb910479d431683101", "status": "in_progress", "external_id": "42", "started_at": "2018-05-04T01:14:52Z", "output": { "title": "Mighty Readme report", "summary": "", "text": "" } }'
```

Response for `in_progress` conclusion

Example response Response schema

Status: 201

```
{ "id": 4, "head_sha": "ce587453ced02b1526dfb4cb910479d431683101", "node_id": "MDg6Q2hlY2tSdW40", "external_id": "42", "url": "https://HOSTNAME/repos/github/hello-world/check-runs/4", "html_url": "https://github.com/github/hello-world/runs/4", "details_url": "https://example.com", "status": "in_progress", "conclusion": null, "started_at": "2018-05-04T01:14:52Z", "completed_at": null, "output": { "title": "Mighty Readme report", "summary": "There are 0 failures, 2 warnings, and 1 notice.", "text": "You may have some misspelled words on lines 2 and 4. You also may want to add a section in your README about how to install your app.", "annotations_count": 2, "annotations_url": "https://HOSTNAME/repos/github/hello-world/check-runs/4/annotations" }, "name": "mighty_readme", "check_suite": { "id":
```

Get a check run

✔ Works with [GitHub Apps](#)

Note: The Checks API only looks for pushes in the repository where the check suite or check run were created. Pushes to a branch in a forked repository are not detected and return an empty `pull_requests` array.

Gets a single check run using its `id`. GitHub Apps must have the `checks:read` permission on a private repository or pull access to a public repository to get check runs. OAuth apps and authenticated users must have the `repo` scope to get check runs in a private repository.

Parameters for "Get a check run"

Headers

`accept` string

Setting to `application/vnd.github+json` is recommended.

Path parameters

`owner` string **Required**

The account owner of the repository. The name is not case sensitive.

`repo` string **Required**

The name of the repository without the `.git` extension. The name is not case sensitive.

`check_run_id` integer **Required**

The unique identifier of the check run.

HTTP response status codes for "Get a check run"

Status code	Description
200	OK

Code samples for "Get a check run"

GET

/repos/{owner}/{repo}/check-runs/{check_run_id}

cURL

JavaScript

GitHub CLI

```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/check-runs/CHECK_RUN_ID
```

Response

Example response

Response schema

Status: 200

```
{ "id": 4, "head_sha": "ce587453ced02b1526dfb4cb910479d431683101", "node_id": "MDg6Q2hLY2tSdW40", "external_id": "", "url": "https://HOSTNAME/repos/github/hello-world/check-runs/4", "html_url": "https://github.com/github/hello-world/runs/4", "details_url": "https://example.com", "status": "completed", "conclusion": "neutral", "started_at": "2018-05-04T01:14:52Z", "completed_at": "2018-05-04T01:14:52Z", "output": { "title": "Mighty Readme report", "summary": "There are 0 failures, 2 warnings, and 1 notice.", "text": "You may have some misspelled words on lines 2 and 4. You also may want to add a section in your README about how to install your app.", "annotations_count": 2, "annotations_url": "https://HOSTNAME/repos/github/hello-world/check-runs/4/annotations" }, "name":
```

Update a check run

✔ Works with [GitHub Apps](#)

Updates a check run for a specific commit in a repository. Your GitHub App must have the `checks:write` permission to edit check runs.

Note: The endpoints to manage checks only look for pushes in the repository where the check suite or check run were created. Pushes to a branch in a forked repository are not detected and return an empty `pull_requests` array.

Parameters for "Update a check run"

Headers

accept string
Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string **Required**
The account owner of the repository. The name is not case sensitive.

repo string **Required**
The name of the repository without the `.git` extension. The name is not case sensitive.

check_run_id integer **Required**
The unique identifier of the check run.

Body parameters

name string
The name of the check. For example, "code-coverage".

details_url string
The URL of the integrator's site that has the full details of the check.

external_id string
A reference for the run on the integrator's system.

started_at string

This is a timestamp in [ISO 8601](#) format: YYYY-MM-DDTHH:MM:SSZ .

status string

The current status.

Can be one of: `queued` , `in_progress` , `completed`

conclusion string

Required if you provide `completed_at` or a `status` of `completed` . The final conclusion of the check. **Note:** Providing `conclusion` will automatically set the `status` parameter to `completed` . You cannot change a check run conclusion to `stale` , only GitHub can set this.

Can be one of: `action_required` , `cancelled` , `failure` , `neutral` , `success` , `skipped` , `stale` , `timed_out`

completed_at string

The time the check completed. This is a timestamp in [ISO 8601](#) format: YYYY-MM-DDTHH:MM:SSZ .

output object

Check runs can accept a variety of data in the `output` object, including a `title` and `summary` and can optionally provide descriptive details about the run.

► Properties of `output`

actions array of objects

Possible further actions the integrator can perform, which a user may trigger. Each action includes a `label` , `identifier` and `description` . A maximum of three actions are accepted. To learn more about check runs and requested actions, see "[Check runs and requested actions](#)."

► Properties of `actions`

HTTP response status codes for "Update a check run"

Status code	Description
200	OK


Code samples for "Update a check run"

PATCH

/repos/{owner}/{repo}/check-runs/{check_run_id}

cURL

JavaScript



```
curl -L \ -X PATCH \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/check-runs/CHECK_RUN_ID \ -d '{"name":"mighty_readme","started_at":"2018-05-04T01:14:52Z","status":"completed","conclusion":"success","completed_at":"2018-05-04T01:14:52Z","output":{"title":"Mighty Readme report","summary":"There are 0 failures, 2 warnings, and 1 notices.","text":"You may have some misspelled words on lines 2 and 4. You also may want to add a section in your README about how to install your app.","annotations":[{"path":"README.md","annotation_level":"warning","title":"Spell Checker","message":"Check your spelling for
```

Response

Example response

Response schema

Status: 200

```
{ "id": 4, "head_sha": "ce587453ced02b1526dfb4cb910479d431683101", "node_id": "MDg6Q2hly2tSdW40", "external_id": "", "url": "https://HOSTNAME/repos/github/hello-world/check-runs/4", "html_url": "https://github.com/github/hello-world/runs/4", "details_url": "https://example.com", "status": "completed", "conclusion": "neutral", "started_at": "2018-05-04T01:14:52Z", "completed_at": "2018-05-04T01:14:52Z", "output": { "title": "Mighty Readme report", "summary": "There are 0 failures, 2 warnings, and 1 notice.", "text": "You may have some misspelled words on lines 2 and 4. You also may want to add a section in your README about how to install your app.", "annotations_count": 2, "annotations_url": "https://HOSTNAME/repos/github/hello-world/check-runs/4/annotations" }, "name":
```

List check run annotations

✔ Works with [GitHub Apps](#)

Lists annotations for a check run using the annotation `id`. GitHub Apps must have the `checks:read` permission on a private repository or pull access to a public repository to get annotations for a check run. OAuth apps and authenticated users must have the `repo` scope to get annotations for a check run in a private repository.

Parameters for "List check run annotations"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string **Required**

The account owner of the repository. The name is not case sensitive.

repo string **Required**

The name of the repository without the `.git` extension. The name is not case sensitive.

check_run_id integer **Required**

The unique identifier of the check run.

Query parameters

per_page integer

The number of results per page (max 100).

Default: `30`

page integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List check run annotations"

Status code	Description
200	OK

Code samples for "List check run annotations"


GET

/repos/{owner}/{repo}/check-runs/{check_run_id}/annotations

cURL

JavaScript

GitHub CLI



```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/check-runs/CHECK_RUN_ID/annotations
```

Response

Example response

Response schema

Status: 200

[{ "path": "README.md", "start_line": 2, "end_line": 2, "start_column": 5, "end_column": 10, "annotation_level": "warning", "title": "Spell Checker", "message": "Check your spelling for 'banaas'.", "raw_details": "Do you mean 'bananas' or 'banana'?", "blob_href": "https://HOSTNAME/repos/github/rest-api-description/git/blobs/abc" }]

Rerequest a check run

✔ Works with [GitHub Apps](#)

Triggers GitHub to rerequest an existing check run, without pushing new code to a repository. This endpoint will trigger the [check_run_webhook](#) event with the action `rerequested`. When a check run is `rerequested`, its `status` is reset to `queued` and the `conclusion` is cleared.

To rerequest a check run, your GitHub App must have the `checks:read` permission on a private repository or pull access to a public repository.

For more information about how to re-run GitHub Actions jobs, see "[Re-run a job from a workflow run](#)".

Parameters for "Rerequest a check run"

Headers

`accept` string
Setting to `application/vnd.github+json` is recommended.

Path parameters

- `owner`

string

Required

The account owner of the repository. The name is not case sensitive.
- `repo`

string

Required

The name of the repository without the `.git` extension. The name is not case sensitive.
- `check_run_id`

integer

Required

The unique identifier of the check run.

HTTP response status codes for "Rerequest a check run"

Status code	Description
201	Created
403	Forbidden if the check run is not rerequestable or doesn't belong to the authenticated GitHub App
404	Resource not found
422	Validation error if the check run is not rerequestable

Code samples for "Rerequest a check run"

POST

/repos/{owner}/{repo}/check-runs/{check_run_id}/rerequest

cURL

JavaScript

GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/check-runs/CHECK_RUN_ID/rerequest
```

Response

Example response

Response schema

Status: 201

List check runs in a check suite

✔ Works with [GitHub Apps](#)

Lists check runs for a check suite using its `id`. GitHub Apps must have the `checks:read` permission on a private repository or pull access to a public repository to get check runs. OAuth apps and authenticated users must have the `repo` scope to get check runs in a private repository.

Note: The endpoints to manage checks only look for pushes in the repository where the check suite or check run were created. Pushes to a branch in a forked repository are not detected and return an empty `pull_requests` array.

Parameters for "List check runs in a check suite"

Headers

accept string
Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string **Required**
The account owner of the repository. The name is not case sensitive.

repo string **Required**
The name of the repository without the `.git` extension. The name is not case sensitive.

check_suite_id integer **Required**
The unique identifier of the check suite.

Query parameters

check_name string
Returns check runs with the specified `name`.

status string

Returns check runs with the specified `status`.

Can be one of: `queued`, `in_progress`, `completed`

filter string

Filters check runs by their `completed_at` timestamp. `latest` returns the most recent check runs.

Default: `latest`

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

per_page integer

The number of results per page (max 100).

Default: `30`

page integer

Page number of the results to fetch.

Default: `1`

HTTP response status codes for "List check runs in a check suite"

Status code	Description
200	OK

Code samples for "List check runs in a check suite"


GET

/repos/{owner}/{repo}/check-suites/{check_suite_id}/check-runs

cURL

JavaScript

GitHub CLI



```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \
http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/check-suites/CHECK_SUITE_ID/check-runs
```

Response

Example response

Response schema

Status: 200

```
{ "total_count": 1, "check_runs": [ { "id": 4, "head_sha": "ce587453ced02b1526dfb4cb910479d431683101", "node_id": "MDg6Q2hlY2tSdW40",
"external_id": "", "url": "https://HOSTNAME/repos/github/hello-world/check-runs/4", "html_url": "https://github.com/github/hello-
world/runs/4", "details_url": "https://example.com", "status": "completed", "conclusion": "neutral", "started_at": "2018-05-04T01:14:52Z",
"completed_at": "2018-05-04T01:14:52Z", "output": { "title": "Mighty Readme report", "summary": "There are 0 failures, 2 warnings, and 1
notice.", "text": "You may have some misspelled words on lines 2 and 4. You also may want to add a section in your README about how to
install your app.", "annotations_count": 2, "annotations_url": "https://HOSTNAME/repos/github/hello-world/check-runs/4/annotations" },
```

List check runs for a Git reference

✔ Works with [GitHub Apps](#)

Lists check runs for a commit ref. The `ref` can be a SHA, branch name, or a tag name. GitHub Apps must have the `checks:read` permission on a private repository or pull access to a public repository to get check runs. OAuth apps and authenticated users must have the `repo` scope to get check runs in a private repository.

Note: The endpoints to manage checks only look for pushes in the repository where the check suite or check run were created. Pushes to a branch in a forked repository are not detected and return an empty `pull_requests` array.

If there are more than 1000 check suites on a single git reference, this endpoint will limit check runs to the 1000 most recent check suites. To iterate over all possible check runs, use the [List check suites for a Git reference](#) endpoint and provide the `check_suite_id` parameter to the [List check runs in a check suite](#) endpoint.

Parameters for "List check runs for a Git reference"

Headers

accept string
Setting to `application/vnd.github+json` is recommended.

Path parameters

owner string **Required**
The account owner of the repository. The name is not case sensitive.

repo string **Required**
The name of the repository without the `.git` extension. The name is not case sensitive.

ref string **Required**
The commit reference. Can be a commit SHA, branch name (`heads/BRANCH_NAME`), or tag name (`tags/TAG_NAME`). For more information, see "[Git References](#)" in the Git documentation.

Query parameters

check_name string
Returns check runs with the specified `name` .

status string
Returns check runs with the specified `status` .
Can be one of: `queued` , `in_progress` , `completed`

filter string
Filters check runs by their `completed_at` timestamp. `latest` returns the most recent check runs.
Default: `latest`
Can be one of: `latest` , `all`

per_page integer
The number of results per page (max 100).
Default: `30`

page integer
Page number of the results to fetch.
Default: `1`

app_id integer

HTTP response status codes for "List check runs for a Git reference"

Status code	Description
-------------	-------------

Code samples for "List check runs for a Git reference"

GET /repos/{owner}/{repo}/commits/{ref}/check-runs

cURL JavaScript GitHub CLI



```
curl -L \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \
http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/commits/REF/check-runs
```

Response

Example response Response schema

Status: 200

```
{ "total_count": 1, "check_runs": [ { "id": 4, "head_sha": "ce587453ced02b1526dfb4cb910479d431683101", "node_id": "MDg6Q2hLY2tSdW40",
"external_id": "", "url": "https://HOSTNAME/repos/github/hello-world/check-runs/4", "html_url": "https://github.com/github/hello-
world/runs/4", "details_url": "https://example.com", "status": "completed", "conclusion": "neutral", "started_at": "2018-05-04T01:14:52Z",
"completed_at": "2018-05-04T01:14:52Z", "output": { "title": "Mighty Readme report", "summary": "There are 0 failures, 2 warnings, and 1
notice.", "text": "You may have some misspelled words on lines 2 and 4. You also may want to add a section in your README about how to
install your app.", "annotations_count": 2, "annotations_url": "https://HOSTNAME/repos/github/hello-world/check-runs/4/annotations" },
```

Legal