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

Code Scanning

Use the REST API to retrieve and update code scanning alerts from a repository.

About code scanning

You can retrieve and update code scanning alerts from a repository. You can use the endpoints to create automated reports for the code scanning alerts in an organization or upload analysis results generated using offline code scanning tools. For more information, see "Finding security vulnerabilities and errors in your code with code scanning."

Custom media type for code scanning

There is one supported custom media type for code scanning endpoints.

```
application/sarif+json
```

You can use this with `GET` requests sent to the `/analyses/{analysis_id}` endpoint. For more information about this operation, see "Get a code scanning analysis for a repository." When you use this media type with this operation, the response includes a subset of the actual data that was uploaded for the specified analysis, rather than the summary of the analysis that's returned when you use the default media type. The response also includes additional data such as the `github/alertNumber` and `github/alertUrl` properties. The data is formatted as SARIF version 2.1.0.

For more information, see "Media types."

List code scanning alerts for an enterprise

Lists code scanning alerts for the default branch for all eligible repositories in an enterprise. Eligible repositories are repositories that are owned by organizations that you own or for which you are a security manager. For more information, see "Managing security managers in your organization."

To use this endpoint, you must be a member of the enterprise, and you must use an access token with the `repo` scope or `security_events` scope.

Parameters for "List code scanning alerts for an enterprise"

Headers

`accept` string

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

Path parameters

`enterprise` string Required

The slug version of the enterprise name. You can also substitute this value with the enterprise id.

Query parameters

`tool_name` string

The name of a code scanning tool. Only results by this tool will be listed. You can specify the tool by using either `tool_name` or `tool_guid`, but not both.

`tool_guid` string or null

The GUID of a code scanning tool. Only results by this tool will be listed. Note that some code scanning tools may not include a GUID in their analysis data. You can specify the tool by using either `tool_guid` or `tool_name`, but not both.

`before` string

A cursor, as given in the Link header. If specified, the query only searches for results before this cursor.

`after` string

A cursor, as given in the Link header. If specified, the query only searches for results after this cursor.

`page` integer

Page number of the results to fetch.

Default: `1`

`per_page` integer

The number of results per page (max 100).

Default: `30`

`direction` string

The direction to sort the results by.

Default: `desc`
Can be one of: `asc`, `desc`

`state` string

If specified, only code scanning alerts with this state will be returned.

Can be one of: `open`, `closed`, `dismissed`, `fixed` `sort` string

The property by which to sort the results.

Default: `created`
Can be one of: `created`, `updated`

HTTP response status codes for "List code scanning alerts for an enterprise"

Status code	Description
`200`	OK
`404`	Resource not found
`503`	Service unavailable

Code samples for "List code scanning alerts for an enterprise"

GET /enterprises/{enterprise}/code-scanning/alerts

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/enterprises/ENTERPRISE/code-scanning/alerts
```

Response

Example response Response schema

Status: 200

```
 [ { "number": 4, "created_at": "2020-02-13T12:29:18Z", "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/alerts/4", "html_url":
"https://github.com/octocat/hello-world/code-scanning/4", "state": "open", "dismissed_by": null, "dismissed_at": null, "dismissed_reason": null, "dismissed_comment":
null, "rule": { "id": "js/zipslip", "severity": "error", "tags": [ "security", "external/cwe/cwe-022" ], "description": "Arbitrary file write during zip extraction",
"name": "js/zipslip" }, "tool": { "name": "CodeQL", "guid": null, "version": "2.4.0" }, "most_recent_instance": { "ref": "refs/heads/main", "analysis_key":
".github/workflows/codeql-analysis.yml:CodeQL-Build", "category": ".github/workflows/codeql-analysis.yml:CodeQL-Build", "environment": "{}", "state": "open",
"commit_sha": "39406e42cb832f683daa691dd652a8dc36ee8930", "message": { "text": "This path depends on a user-provided value." }, "location": { "path": "spec-main/api-
```

List code scanning alerts for an organization

✓ Works with [GitHub Apps](#)

Lists code scanning alerts for the default branch for all eligible repositories in an organization. Eligible repositories are repositories that are owned by organizations that you own or for which you are a security manager. For more information, see "[Managing security managers in your organization](#)."

To use this endpoint, you must be an owner or security manager for the organization, and you must use an access token with the `repo` scope or `security_events` scope.

For public repositories, you may instead use the `public_repo` scope.

GitHub Apps must have the `security_events` read permission to use this endpoint.

Parameters for "List code scanning alerts for an organization"

Headers

`accept` string

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

Path parameters

`org` string Required

The organization name. The name is not case sensitive.

Query parameters

`tool_name` string

The name of a code scanning tool. Only results by this tool will be listed. You can specify the tool by using either `tool_name` or `tool_guid`, but not both.

`tool_guid` string or null

The GUID of a code scanning tool. Only results by this tool will be listed. Note that some code scanning tools may not include a GUID in their analysis data. You can specify the tool by using either `tool_guid` or `tool_name`, but not both.

`before` string

A cursor, as given in the [Link header](#). If specified, the query only searches for results before this cursor.

`after` string

A cursor, as given in the [Link header](#). If specified, the query only searches for results after this cursor.

`page` integer

Page number of the results to fetch.

Default: `1`

`per_page` integer

The number of results per page (max 100).

Default: `30`

`direction` string

The direction to sort the results by.

Default: `desc`
Can be one of: `asc`, `desc`

`state` string

If specified, only code scanning alerts with this state will be returned.

Can be one of: `open`, `closed`, `dismissed`, `fixed`

`sort` string

The property by which to sort the results.

Default: `created`
Can be one of: `created`, `updated`

`severity` string

If specified, only code scanning alerts with this severity will be returned.

Can be one of: `critical`, `high`, `medium`, `low`, `warning`, `note`, `error`

HTTP response status codes for "List code scanning alerts for an organization"

Status code	Description
`200`	OK
`404`	Resource not found
`503`	Service unavailable

Code samples for "List code scanning alerts for an organization"

GET /orgs/{org}/code-scanning/alerts

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/orgs/ORG/code-scanning/alerts
```

Response

Example response Response schema

```
Status: 200
```

[{ "number": 4, "created_at": "2020-02-13T12:29:18Z", "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/alerts/4", "html_url": "https://github.com/octocat/hello-world/code-scanning/4", "state": "open", "dismissed_by": null, "dismissed_at": null, "dismissed_reason": null, "dismissed_comment": null, "rule": { "id": "js/zipslip", "severity": "error", "tags": ["security", "external/cwe/cwe-022"], "description": "Arbitrary file write during zip extraction", "name": "js/zipslip" }, "tool": { "name": "CodeQL", "guid": null, "version": "2.4.0" }, "most_recent_instance": { "ref": "refs/heads/main", "analysis_key": ".github/workflows/codeql-analysis.yml:CodeQL-Build", "category": ".github/workflows/codeql-analysis.yml:CodeQL-Build", "environment": "{}", "state": "open", "commit_sha": "39406e42cb832f683daa691dd652a8dc36ee8930", "message": { "text": "This path depends on a user-provided value." }, "location": { "path": "spec-main/api-

List code scanning alerts for a repository

✓ Works with GitHub Apps

Lists code scanning alerts.

To use this endpoint, you must use an access token with the `security_events` scope or, for alerts from public repositories only, an access token with the `public_repo` scope.

GitHub Apps must have the `security_events` read permission to use this endpoint.

The response includes a `most_recent_instance` object. This provides details of the most recent instance of this alert for the default branch (or for the specified Git reference if you used `ref` in the request).

Parameters for "List code scanning alerts for a repository"

Headers

`accept` string

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

Path parameters

`owner` string Required

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

`repo` string Required

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

Query parameters

`tool_name` string

The name of a code scanning tool. Only results by this tool will be listed. You can specify the tool by using either `tool_name` or `tool_guid`, but not both.

`tool_guid` string or null

The GUID of a code scanning tool. Only results by this tool will be listed. Note that some code scanning tools may not include a GUID in their analysis data. You can specify the tool by using either `tool_guid` or `tool_name`, but not both.

`page` integer

Page number of the results to fetch.

Default: `1`

`per_page` integer

The number of results per page (max 100).

Default: `30`

`ref` string

The Git reference for the results you want to list. The `ref` for a branch can be formatted either as `refs/heads/<branch name>` or simply `<branch name>`. To reference a pull request use `refs/pull/<number>/merge`.

`direction` string

The direction to sort the results by.

Default: `desc`
Can be one of: `asc`, `desc`

`sort` string

The property by which to sort the results.

Default: `created`
Can be one of: `created`, `updated`

`state` string

If specified, only code scanning alerts with this state will be returned.

Can be one of: `open`, `closed`, `dismissed`, `fixed`

`severity` string

If specified, only code scanning alerts with this severity will be returned.

Can be one of: `critical`, `high`, `medium`, `low`, `warning`, `note`, `error`

HTTP response status codes for "List code scanning alerts for a repository"

Status code	Description
`200`	OK
`304`	Not modified
`403`	Response if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "List code scanning alerts for a repository"

GET `/repos/{owner}/{repo}/code-scanning/alerts`

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/code-scanning/alerts
```

Response

Example response Response schema

```
Status: 200
```

```
[ { "number": 4, "created_at": "2020-02-13T12:29:18Z", "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/alerts/4", "html_url":
"https://github.com/octocat/hello-world/code-scanning/4", "state": "open", "fixed_at": null, "dismissed_by": null, "dismissed_at": null, "dismissed_reason": null,
"dismissed_comment": null, "rule": { "id": "js/zipslip", "severity": "error", "tags": [ "security", "external/cwe/cwe-022" ], "description": "Arbitrary file write during
zip extraction", "name": "js/zipslip" }, "tool": { "name": "CodeQL", "guid": null, "version": "2.4.0" }, "most_recent_instance": { "ref": "refs/heads/main",
"analysis_key": ".github/workflows/codeql-analysis.yml:CodeQL-Build", "category": ".github/workflows/codeql-analysis.yml:CodeQL-Build", "environment": "{}", "state":
"open", "commit_sha": "39406e42cb832f683daa691dd652a8dc36ee8930", "message": { "text": "This path depends on a user-provided value." }, "location": { "path": "spec-
```

Get a code scanning alert

✓ Works with GitHub Apps

Gets a single code scanning alert. You must use an access token with the `security_events` scope to use this endpoint with private repos, the `public_repo` scope also grants permission to read security events on public repos only. GitHub Apps must have the `security_events` read permission to use this endpoint.

Parameters for "Get a code scanning alert"

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.

`alert_number` integer Required

The number that identifies an alert. You can find this at the end of the URL for a code scanning alert within GitHub, and in the `number` field in the response from the `GET /repos/{owner}/{repo}/code-scanning/alerts` operation.

HTTP response status codes for "Get a code scanning alert"

Status code	Description
`200`	OK
`304`	Not modified
`403`	Response if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "Get a code scanning alert"

GET /repos/{owner}/{repo}/code-scanning/alerts/{alert_number}

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/code-scanning/alerts/ALERT_NUMBER
```

Response

Example response Response schema

```
Status: 200
```

```
 { "number": 42, "created_at": "2020-06-19T11:21:34Z", "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/alerts/42", "html_url":
"https://github.com/octocat/hello-world/code-scanning/42", "state": "dismissed", "fixed_at": null, "dismissed_by": { "login": "octocat", "id": 54933897, "node_id":
"MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif", "gravatar_id": "", "url": "https://HOSTNAME/users/octocat", "html_url":
"https://github.com/octocat", "followers_url": "https://HOSTNAME/users/octocat/followers", "following_url": "https://HOSTNAME/users/octocat/following{/other_user}",
"gists_url": "https://HOSTNAME/users/octocat/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
"https://HOSTNAME/users/octocat/subscriptions", "organizations_url": "https://HOSTNAME/users/octocat/orgs", "repos_url": "https://HOSTNAME/users/octocat/repos",
```

Update a code scanning alert

✓ Works with GitHub Apps

Updates the status of a single code scanning alert. You must use an access token with the `security_events` scope to use this endpoint with private repositories. You can also use tokens with the `public_repo` scope for public repositories only. GitHub Apps must have the `security_events` write permission to use this endpoint.

Parameters for "Update a code scanning alert"

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.

`alert_number` integer Required

The number that identifies an alert. You can find this at the end of the URL for a code scanning alert within GitHub, and in the `number` field in the response from the `GET /repos/{owner}/{repo}/code-scanning/alerts` operation.

Body parameters

`state` string Required

Sets the state of the code scanning alert. You must provide `dismissed_reason` when you set the state to `dismissed`.

Can be one of: `open`, `dismissed`

`dismissed_reason` string or null

Required when the state is dismissed. The reason for dismissing or closing the alert.

Can be one of: *`null`*, `false positive`, `won't fix`, `used in tests`

`dismissed_comment` string or null

The dismissal comment associated with the dismissal of the alert.

HTTP response status codes for "Update a code scanning alert"

Status code	Description
`200`	OK
`403`	Response if the repository is archived or if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "Update a code scanning alert"

PATCH `/repos/{owner}/{repo}/code-scanning/alerts/{alert_number}`

cURL JavaScript GitHub CLI

```
 curl -L \ -X PATCH \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/code-scanning/alerts/ALERT_NUMBER \ -d '{"state":"dismissed","dismissed_reason":"false positive","dismissed_comment":"This alert
is not actually correct, because there'\''s a sanitizer included in the library."}'
```

Response

Example response Response schema

```
Status: 200
```

{ "number": 42, "created_at": "2020-08-25T21:28:36Z", "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/alerts/42", "html_url":
"https://github.com/octocat/hello-world/code-scanning/42", "state": "dismissed", "fixed_at": null, "dismissed_by": { "login": "octocat", "id": 1, "node_id":
"MDQ6VXNlcjE=", "avatar_url": "https://github.com/images/error/octocat_happy.gif", "gravatar_id": "", "url": "https://HOSTNAME/users/octocat", "html_url":
"https://github.com/octocat", "followers_url": "https://HOSTNAME/users/octocat/followers", "following_url": "https://HOSTNAME/users/octocat/following{/other_user}",
"gists_url": "https://HOSTNAME/users/octocat/gists{/gist_id}", "starred_url": "https://HOSTNAME/users/octocat/starred{/owner}{/repo}", "subscriptions_url":
"https://HOSTNAME/users/octocat/subscriptions", "organizations_url": "https://HOSTNAME/users/octocat/orgs", "repos_url": "https://HOSTNAME/users/octocat/repos",

List instances of a code scanning alert

✓ Works with GitHub Apps

Lists all instances of the specified code scanning alert. You must use an access token with the `security_events` scope to use this endpoint with private repos, the `public_repo` scope also grants permission to read security events on public repos only. GitHub Apps must have the `security_events` read permission to use this endpoint.

Parameters for "List instances of a code scanning alert"

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.

`alert_number` integer Required

The number that identifies an alert. You can find this at the end of the URL for a code scanning alert within GitHub, and in the `number` field in the response from the `GET /repos/{owner}/{repo}/code-scanning/alerts` operation.

Query parameters

`page` integer

Page number of the results to fetch.

Default: `1`

`per_page` integer

The number of results per page (max 100).

Default: `30`

`ref` string

The Git reference for the results you want to list. The `ref` for a branch can be formatted either as `refs/heads/<branch name>` or simply `<branch name>`. To reference a pull request use `refs/pull/<number>/merge`.

HTTP response status codes for "List instances of a code scanning alert"

Status code	Description
`200`	OK
`403`	Response if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "List instances of a code scanning alert"

GET `/repos/{owner}/{repo}/code-scanning/alerts/{alert_number}/instances`

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/code-scanning/alerts/ALERT_NUMBER/instances
```

Response

Example response Response schema

```
Status: 200
```

```
 [ { "ref": "refs/heads/main", "analysis_key": ".github/workflows/codeql-analysis.yml:CodeQL-Build", "environment": "", "category": ".github/workflows/codeql-
analysis.yml:CodeQL-Build", "state": "open", "fixed_at": null, "commit_sha": "39406e42cb832f683daa691dd652a8dc36ee8930", "message": { "text": "This path depends on a
user-provided value." }, "location": { "path": "lib/ab12-gen.js", "start_line": 917, "end_line": 917, "start_column": 7, "end_column": 18 }, "classifications": [
"library" ] }, { "ref": "refs/pull/3740/merge", "analysis_key": ".github/workflows/codeql-analysis.yml:CodeQL-Build", "environment": "", "category":
".github/workflows/codeql-analysis.yml:CodeQL-Build", "state": "fixed", "fixed_at": "2020-02-14T12:29:18Z", "commit_sha": "b09da05606e27f463a2b49287684b4ae777092f2",
"message": { "text": "This suffix check is missing a length comparison to correctly handle lastIndexOf returning -1." }, "location": { "path": "app/script.js",
```

List code scanning analyses for a repository

Works with GitHub Apps

Lists the details of all code scanning analyses for a repository, starting with the most recent. The response is paginated and you can use the `page` and `per_page` parameters to list the analyses you're interested in. By default 30 analyses are listed per page.

The `rules_count` field in the response give the number of rules that were run in the analysis. For very old analyses this data is not available, and `0` is returned in this field.

You must use an access token with the `security_events` scope to use this endpoint with private repos, the `public_repo` scope also grants permission to read security events on public repos only. GitHub Apps must have the `security_events` read permission to use this endpoint.

Deprecation notice: The `tool_name` field is deprecated and will, in future, not be included in the response for this endpoint. The example response reflects this change. The tool name can now be found inside the `tool` field.

Parameters for "List code scanning analyses for a repository"

Headers

`accept` string

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

Path parameters

`owner` string Required

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

`repo` string Required

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

Query parameters

`tool_name` string

The name of a code scanning tool. Only results by this tool will be listed. You can specify the tool by using either `tool_name` or `tool_guid`, but not both.

`tool_guid` string or null

The GUID of a code scanning tool. Only results by this tool will be listed. Note that some code scanning tools may not include a GUID in their analysis data. You can specify the tool by using either `tool_guid` or `tool_name`, but not both.

page integer

Page number of the results to fetch.

Default: `1`

per_page integer

The number of results per page (max 100).

Default: `30`

ref string

The Git reference for the analyses you want to list. The `ref` for a branch can be formatted either as `refs/heads/<branch name>` or simply `<branch name>`. To reference a pull request use `refs/pull/<number>/merge`.

sarif_id string

Filter analyses belonging to the same SARIF upload.

direction string

The direction to sort the results by.

Default: `desc`
Can be one of: `asc`, `desc`

sort string

The property by which to sort the results.

Default: `created`
Value: `created`

HTTP response status codes for "List code scanning analyses for a repository"

Status code	Description
`200`	OK
`403`	Response if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "List code scanning analyses for a repository"

GET `/repos/{owner}/{repo}/code-scanning/analyses`

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/code-scanning/analyses
```

Response

Example response Response schema

```
Status: 200
```

```
 [ { "ref": "refs/heads/main", "commit_sha": "d99612c3e1f2970085cfbaeadf8f010ef69bad83", "analysis_key": ".github/workflows/codeql-analysis.yml:analyze", "environment": "
{\"language\":\"python\"}", "error": "", "category": ".github/workflows/codeql-analysis.yml:analyze/language:python", "created_at": "2020-08-27T15:05:21Z",
"results_count": 17, "rules_count": 49, "id": 201, "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/analyses/201", "sarif_id": "6c81cd8e-b078-4ac3-a3be-
1dad7dbd0b53", "tool": { "name": "CodeQL", "guid": null, "version": "2.4.0" }, "deletable": true, "warning": "" }, { "ref": "refs/heads/my-branch", "commit_sha":
"c8cff6510d4d084fb1b4aa13b64b97ca12b07321", "analysis_key": ".github/workflows/shiftleft.yml:build", "environment": "{}", "error": "", "category":
".github/workflows/shiftleft.yml:build/", "created_at": "2020-08-31T22:46:44Z", "results_count": 17, "rules_count": 32, "id": 200, "url":
```

Get a code scanning analysis for a repository

✓ Works with GitHub Apps

Gets a specified code scanning analysis for a repository. You must use an access token with the `security_events` scope to use this endpoint with private repos, the `public_repo` scope also grants permission to read security events on public repos only. GitHub Apps must have the `security_events` read permission to use this endpoint.

The default JSON response contains fields that describe the analysis. This includes the Git reference and commit SHA to which the analysis relates, the datetime of the analysis, the name of the code scanning tool, and the number of alerts.

The `rules_count` field in the default response give the number of rules that were run in the analysis. For very old analyses this data is not available, and `0` is returned in this field.

If you use the Accept header `application/sarif+json`, the response contains the analysis data that was uploaded. This is formatted as SARIF version 2.1.0.

Parameters for "Get a code scanning analysis for a repository"

Headers

accept string

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

Path parameters

`owner` string Required

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

`repo` string Required

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

`analysis_id` integer Required

The ID of the analysis, as returned from the `GET /repos/{owner}/{repo}/code-scanning/analyses` operation.

HTTP response status codes for "Get a code scanning analysis for a repository"

Status code	Description
`200`	OK
`403`	Response if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "Get a code scanning analysis for a repository"

Example 1: Status Code 200 (application/json)

GET `/repos/{owner}/{repo}/code-scanning/analyses/{analysis_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/code-scanning/analyses/ANALYSIS_ID
```

application/json response

Example response Response schema

```
Status: 200
```

```
 { "ref": "refs/heads/main", "commit_sha": "c18c69115654ff0166991962832dc2bd7756e655", "analysis_key": ".github/workflows/codeql-analysis.yml:analyze", "environment": "
{\"language\":\"javascript\"}", "error": "", "category": ".github/workflows/codeql-analysis.yml:analyze/language:javascript", "created_at": "2021-01-13T11:55:49Z",
"results_count": 3, "rules_count": 67, "id": 3602840, "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/analyses/201", "sarif_id": "47177e22-5596-11eb-80a1-
c1e54ef945c6", "tool": { "name": "CodeQL", "guid": null, "version": "2.4.0" }, "deletable": true, "warning": "" }
```

Delete a code scanning analysis from a repository

✓ Works with GitHub Apps

Deletes a specified code scanning analysis from a repository. For private repositories, you must use an access token with the `repo` scope. For public repositories, you must use an access token with `public_repo` scope. GitHub Apps must have the `security_events` write permission to use this endpoint.

You can delete one analysis at a time. To delete a series of analyses, start with the most recent analysis and work backwards. Conceptually, the process is similar to the undo function in a text editor.

When you list the analyses for a repository, one or more will be identified as deletable in the response:

```
"deletable": true
```

An analysis is deletable when it's the most recent in a set of analyses. Typically, a repository will have multiple sets of analyses for each enabled code scanning tool, where a set is determined by a unique combination of analysis values:

- `ref`
- `tool`
- `category`

If you attempt to delete an analysis that is not the most recent in a set, you'll get a 400 response with the message:

```
Analysis specified is not deletable.
```

The response from a successful `DELETE` operation provides you with two alternative URLs for deleting the next analysis in the set: `next_analysis_url` and `confirm_delete_url`. Use the `next_analysis_url` URL if you want to avoid accidentally deleting the final analysis in a set. This is a useful option if you want to preserve at least one analysis for the specified tool in your repository. Use the `confirm_delete_url` URL if you are content to remove all analyses for a tool. When you delete the last analysis in a set, the value of `next_analysis_url` and `confirm_delete_url` in the 200 response is `null`.

As an example of the deletion process, let's imagine that you added a workflow that configured a particular code scanning tool to analyze the code in a repository. This tool has added 15 analyses: 10 on the default branch, and another 5 on a topic branch. You therefore have two separate sets of analyses for this tool. You've now decided that you want to remove all of the analyses for the tool. To do this you must make 15 separate deletion requests. To start, you must find an analysis that's identified as deletable. Each set of analyses always has one that's identified as deletable. Having found the deletable analysis for one of the two sets, delete this analysis and then continue deleting the next analysis in the set until they're all deleted. Then repeat the process for the second set. The procedure therefore consists of a nested loop:

Outer loop:

- List the analyses for the repository, filtered by tool.

- Parse this list to find a deletable analysis. If found:

  Inner loop:

  - Delete the identified analysis.
  - Parse the response for the value of `confirm_delete_url` and, if found, use this in the next iteration.

The above process assumes that you want to remove all trace of the tool's analyses from the GitHub user interface, for the specified repository, and it therefore uses the `confirm_delete_url` value. Alternatively, you could use the `next_analysis_url` value, which would leave the last analysis in each set undeleted to avoid removing a tool's analysis entirely.

Parameters for "Delete a code scanning analysis from a repository"

Headers

`accept` string

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

Path parameters

`owner` string Required

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

`repo` string Required

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

`analysis_id` integer Required

The ID of the analysis, as returned from the `GET /repos/{owner}/{repo}/code-scanning/analyses` operation.

Query parameters

`confirm_delete` string or null

Allow deletion if the specified analysis is the last in a set. If you attempt to delete the final analysis in a set without setting this parameter to `true`, you'll get a 400 response with the message: `Analysis is last of its type and deletion may result in the loss of historical alert data. Please specify confirm_delete.`

HTTP response status codes for "Delete a code scanning analysis from a repository"

Status code	Description
`200`	OK
`400`	Bad Request
`403`	Response if the repository is archived or if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "Delete a code scanning analysis from a repository"

DELETE /repos/{owner}/{repo}/code-scanning/analyses/{analysis_id}

cURL JavaScript GitHub CLI

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/code-scanning/analyses/ANALYSIS_ID
```

Default response

Example response Response schema

```
Status: 200
```

```
{ "next_analysis_url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/analyses/41", "confirm_delete_url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/analyses/41?confirm_delete" }
```

Get a code scanning default setup configuration

✓ Works with GitHub Apps

Gets a code scanning default setup configuration. You must use an access token with the `repo` scope to use this endpoint with private repos or the `public_repo` scope for public repos. GitHub Apps must have the `repo` write permission to use this endpoint.

Parameters for "Get a code scanning default setup configuration"

Headers

`accept` string

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

Path parameters

`owner` string Required

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

`repo` string Required

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

HTTP response status codes for "Get a code scanning default setup configuration"

Status code	Description
`200`	OK
`403`	Response if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`503`	Service unavailable

Code samples for "Get a code scanning default setup configuration"

GET `/repos/{owner}/{repo}/code-scanning/default-setup`

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/code-scanning/default-setup
```

Response

Example response Response schema

`Status: 200`

```
{ "state": "configured", "languages": [ "ruby", "python" ], "query_suite": "default", "updated_at": "2023-01-19T11:21:34Z" }
```

Update a code scanning default setup configuration

✓ Works with GitHub Apps

Updates a code scanning default setup configuration. You must use an access token with the `repo` scope to use this endpoint with private repos or the `public_repo` scope for public repos. GitHub Apps must have the `repo` write permission to use this endpoint.

Parameters for "Update a code scanning default setup configuration"

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

`state` string Required

Whether code scanning default setup has been configured or not.

Can be one of: `configured`, `not-configured`

`query_suite` string

CodeQL query suite to be used.

Can be one of: `default`, `extended`

`languages` array of strings

CodeQL languages to be analyzed. Supported values are: `c-cpp`, `csharp`, `go`, `java-kotlin`, `javascript-typescript`, `python`, `ruby`

HTTP response status codes for "Update a code scanning default setup configuration"

Status code	Description
`200`	OK
`202`	Accepted
`403`	Response if the repository is archived or if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`409`	Response if there is already a validation run in progress with a different default setup configuration
`503`	Service unavailable

Code samples for "Update a code scanning default setup configuration"

PATCH /repos/{owner}/{repo}/code-scanning/default-setup

cURL JavaScript GitHub CLI

```
curl -L \ -X PATCH \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
http(s)://HOSTNAME/api/v3/repos/OWNER/REPO/code-scanning/default-setup \ -d '{"state":"configured"}'
```

Response

Example response Response schema

Status: 202

```
{ "run_id": 42, "run_url": "https://HOSTNAME/repos/octoorg/octocat/actions/runs/42" }
```

Upload an analysis as SARIF data

✓ Works with GitHub Apps

Uploads SARIF data containing the results of a code scanning analysis to make the results available in a repository. You must use an access token with the `security_events` scope to use this endpoint for private repositories. You can also use tokens with the `public_repo` scope for public repositories only. GitHub Apps must have the `security_events` write permission to use this endpoint. For troubleshooting information, see "Troubleshooting SARIF uploads."

There are two places where you can upload code scanning results.

- If you upload to a pull request, for example `--ref refs/pull/42/merge` or `--ref refs/pull/42/head`, then the results appear as alerts in a pull request check. For more information, see "Triaging code scanning alerts in pull requests."
- If you upload to a branch, for example `--ref refs/heads/my-branch`, then the results appear in the **Security** tab for your repository. For more information, see "Managing code scanning alerts for your repository."

You must compress the SARIF-formatted analysis data that you want to upload, using `gzip`, and then encode it as a Base64 format string. For example:

```
gzip -c analysis-data.sarif | base64 -w0
```

SARIF upload supports a maximum number of entries per the following data objects, and an analysis will be rejected if any of these objects is above its maximum value. For some objects, there are additional values over which the entries will be ignored while keeping the most important entries whenever applicable. To get the most out of your analysis when it includes data above the supported limits, try to optimize the analysis configuration. For example, for the CodeQL tool, identify and remove the most noisy queries. For more information, see "[SARIF results exceed one or more limits](https://docs.github.com/enterprise-server@3.10/code-security/code-scanning/troubleshooting-sarif/results-exceed-limit)."

SARIF data	Maximum values	Additional limits
Runs per file	20	
Results per run	25,000	Only the top 5,000 results will be included, prioritized by severity.
Rules per run	25,000	
Tool extensions per run	100	
Thread Flow Locations per result	10,000	Only the top 1,000 Thread Flow Locations will be included, using prioritization.
Location per result	1,000	Only 100 locations will be included.
Tags per rule	20	Only 10 tags will be included.

The `202 Accepted` response includes an `id` value. You can use this ID to check the status of the upload by using it in the `/sarifs/{sarif_id}` endpoint. For more information, see "Get information about a SARIF upload."

Parameters for "Upload an analysis as SARIF data"

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

`commit_sha` string Required

The SHA of the commit to which the analysis you are uploading relates.

`ref` string Required

The full Git reference, formatted as `refs/heads/<branch name>`, `refs/pull/<number>/merge`, or `refs/pull/<number>/head`.

`sarif` string Required

A Base64 string representing the SARIF file to upload. You must first compress your SARIF file using `gzip` and then translate the contents of the file into a Base64 encoding string. For more information, see "SARIF support for code scanning."

`checkout_uri` string

The base directory used in the analysis, as it appears in the SARIF file. This property is used to convert file paths from absolute to relative, so that alerts can be mapped to their correct location in the repository.

`started_at` string

The time that the analysis run began. This is a timestamp in ISO 8601 format: `YYYY-MM-DDTHH:MM:SSZ`.

`tool_name` string

The name of the tool used to generate the code scanning analysis. If this parameter is not used, the tool name defaults to "API". If the uploaded SARIF contains a tool GUID, this will be available for filtering using the `tool_guid` parameter of operations such as `GET /repos/{owner}/{repo}/code-scanning/alerts`.

HTTP response status codes for "Upload an analysis as SARIF data"

Status code	Description
`202`	Accepted
`400`	Bad Request if the sarif field is invalid
`403`	Response if the repository is archived or if GitHub Advanced Security is not enabled for this repository
`404`	Resource not found
`413`	Payload Too Large if the sarif field is too large
`503`	Service unavailable

Code samples for "Upload an analysis as SARIF data"

POST `/repos/{owner}/{repo}/code-scanning/sarifs`

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/code-sca
'{"commit_sha":"4b6472266afd7b471e86085a6659e8c7f2b119da","ref":"refs/heads/master","sarif":"H4sICMLGdF4AA2V4YW1wbGUuc2FyaWYAvVjdbts2FL7PUxDCijaA/CM7iRNfLkPXYgHSNstumlzQ0pHFVCI1korjFgH2ONtr7Ul2KFmy
```

Default response

Example response Response schema

Status: 202

```
{ "id": "47177e22-5596-11eb-80a1-c1e54ef945c6", "url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/sarifs/47177e22-5596-11eb-80a1-c1e54ef945c6" }
```

Get information about a SARIF upload

✓ Works with GitHub Apps

Gets information about a SARIF upload, including the status and the URL of the analysis that was uploaded so that you can retrieve details of the analysis. For more information, see "Get a code scanning analysis for a repository." You must use an access token with the `security_events` scope to use this endpoint with private repos, the `public_repo` scope also grants permission to read security events on public repos only. GitHub Apps must have the `security_events` read permission to use this endpoint.

Parameters for "Get information about a SARIF upload"

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.

`sarif_id` string Required

The SARIF ID obtained after uploading.

HTTP response status codes for "Get information about a SARIF upload"

Status code	Description
`200`	OK
`403`	Response if GitHub Advanced Security is not enabled for this repository
`404`	Not Found if the sarif id does not match any upload
`503`	Service unavailable

Code samples for "Get information about a SARIF upload"

GET `/repos/{owner}/{repo}/code-scanning/sarifs/{sarif_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/code-scanning/sarifs/SARIF_ID
```

Default response

Example response Response schema

```
Status: 200
```

```
{ "processing_status": "complete", "analyses_url": "https://HOSTNAME/repos/octocat/hello-world/code-scanning/analyses?sarif_id=47177e22-5596-11eb-80a1-c1e54ef945c6" }
```