Q



REST API / Repositories / Rule Suites

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

Rule Suites

Use the REST API to manage rule suites for repositories.

Note: The API endpoints for rule suites are in public beta and subject to change.

List repository rule suites *₽*

Works with GitHub Apps

Lists suites of rule evaluations at the repository level. For more information, see "Managing rulesets for a repository."

Parameters for "List repository rule suites"

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

ref string

The name of the ref. Cannot contain wildcard characters. When specified, only rule evaluations triggered for this ref will be returned.

time_period string

The time period to filter by.

For example, day will filter for rule suites that occurred in the past 24 hours, and week will filter for insights that occurred in the past 7 days (168 hours).

Default: day

Can be one of: hour , day , week , month

actor_name string

The handle for the GitHub user account to filter on. When specified, only rule evaluations triggered by this actor will be returned.

rule_suite_result string

The rule results to filter on. When specified, only suites with this result will be returned.

Default: all

Can be one of: pass, fail, bypass, 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 repository rule suites"

Status code	Description
200	OK
404	Resource not found
500	Internal Error

Code samples for "List repository rule suites"



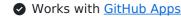
Response

Example response Response schema

Status: 200

[{ "id": 21, "actor_id": 12, "username": "octocat", "before_sha": "893f768e172fb1bc9c5d6f3dd48557e45f14e01d", "after_sha": "dedd88641a362b6b4ea872da4847d6131a164d01", "ref": "refs/heads/i-see-everything", "repository_id": 404, "repository_name": "octorepo", "pushed_at": "2023-07-06T08:43:03Z", "result": "bypass" }, { "id": 25, "actor_id": 11, "username": "not-octocat", "before_sha": "48994e4e01ccc943624c6231f172702b82b233cc", "after_sha": "ecfd5a1025fa271a33ca5608d089476a2df3c9a1", "ref": "refs/heads/i-am-everything", "repository_id": 404, "repository_name": "octo-repo", "pushed_at": "2023-07-07T08:43:03Z", "result": "pass", "evaluation_result": "fail" }]

Get a repository rule suite *∂*



Gets information about a suite of rule evaluations from within a repository. For more information, see "Managing rulesets for a repository."

Parameters for "Get a repository rule 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.

rule_suite_id integer Required

The unique identifier of the rule suite result. To get this ID, you can use <u>GET /repos/{owner}/{repo}/rulesets/rule-suites</u> for repositories and <u>GET /orgs/{org}/rulesets/rule-suites</u> for organizations.

HTTP response status codes for "Get a repository rule suite"

Status code	Description
200	OK
404	Resource not found
500	Internal Error

Code samples for "Get a repository rule suite"



Response

Example response Response schema

Status: 200

{ "id": 21, "actor_id": 12, "username": "octocat", "before_sha": "893f768e172fb1bc9c5d6f3dd48557e45f14e01d", "after_sha": "dedd88641a362b6b4ea872da4847d6131a164d01", "ref": "refs/heads/i-see-everything", "repository_id": 404, "repository_name": "octorepo", "pushed_at": "2023-07-06T08:43:03Z", "result": "bypass", "evaluation_result": "fail", "rule_evaluations": [{ "rule_source": { "type": "ruleset", "id": 2, "name": "Author email must be a GitHub email address" }, "enforcement": "active", "result": "pass", "rule_type": "commit_author_email_pattern" }, { "rule_source": { "type": "protected_branch" }, "enforcement": "active", "result": "fail". "rule_type": "pull_request". "details": "Changes_must_be_made_through_a_pull_request." }, { "rule_source": { "type": "rule_source": { "type": "protected_branch" }, "enforcement": "active", "result": "fail". "rule_type": "pull_request". "details": "Changes_must_be_made_through_a_pull_request." }, { "rule_source": { "type": "protected_branch" }, "rule_source": { "type": "protected_branch" }, "enforcement": "active", "result": "fail". "rule_type": "postected_branch" }, "enforcement": "acti

rate) race_cype i pace_request) accases i changes mast be made chroagh a pace request. [] [race_source i [cype i

Legal

© 2023 GitHub, Inc. <u>Terms</u> <u>Privacy</u> <u>Status</u> <u>Pricing</u> <u>Expert services</u> <u>Blog</u>