

PR #2676 完整报告

radixark/miles

feat(ci): acknowledge rerun-test commands

合并时间: 2026-08-21 12:34

原文链接: <http://prhub.com.cn/radixark/miles/pull/2676>

执行摘要

- 一句话: 为 /rerun-test 增加 +1 确认与 Actions 运行链接回复
- 推荐动作: 建议 CI 平台方向工程师精读。值得关注的设计: dispatch 时直接请求 run details 避免轮询、以 success_reaction 声明式扩展反馈、反馈 job 与 dispatch job 的 token 权限隔离。对普通工程师, 体感是 /rerun-test 有了明确确认与直达链接, 调试效率提升明显。整体改动小而完整, 可作为 GitHub Actions 命令网关的参考范例。

功能与动机

PR body 的 Motivation 明确说明: 接受的目标化测试请求需要从 PR 对话直接到达其 Actions run, 要求“The PR feedback must confirm acceptance and identify the exact run without requiring a search through the Actions list”。此前 /rerun-test 触发后没有任何反馈, 用户只能去 Actions 列表人工找 run; 本 PR 参考 SGLang 的 reaction + workflow-link 回复模型, 补上“确认接受 + 直达链接”的闭环, 但刻意省略 SGLang 后续的 running/pass/fail 评论编辑。

实现拆解

1. 扩充命令注册表: 在 `.github/workflows/scripts/comment_ci_command.py` 中为 `CommandSpec` 增加 `success_reaction` 字段, 静态注册表仅把 `RunTestFile` 标记为 +1, 其余命令为 none; `.github/workflows/comment-ci-command.yml` 的 preflight 校验 `success_reaction` 只能取 none 或 +1, 从源头限制反馈行为。
2. 改造 dispatch 并收紧校验: `GitHubAPI.create_workflow_dispatch` 在 payload 中携带 `return_run_details: true`, 期望状态码从 204 改为 200; `_request` 从单值 `expected_status` 改为 `expected_statuses` 元组, 以兼容不同端点的 200/201/204 响应。返回体经 `_positive_int` 校验 `workflow_run_id`、`_validate_workflow_run_url` 校验 `html_url` 后, 只导出该 URL 供后续反馈使用, 全程不轮询、不执行 PR 代码。
3. 新增两个独立反馈 job: `comment-ci-command.yml` 新增 `acknowledge-command` (dispatch 成功后用仅 `Issues: write` 的 App token 打 +1) 与 `reply-command` (读取 `actions-command` 输出的 `workflow_run_url`, 发布 `[View workflow run](...)` 链接回复)。两个 job 都只 checkout 受信任的 handler 文件, 反馈失败不会再次 dispatch 测试。
4. 测试与文档配套: `tests/ci/test/test_comment_ci_command.py` 新增约 440 行, 覆盖 dispatch 端点与 run-detail 校验 (含 204 空响应拒绝)、URL 验证与输出传播、+1 与链接回复发布、token/job 隔离; `docs/ci/01-label.md` 同步修正命令行为描述与过时迁移文案。

本地 227 个测试通过，pre-commit 全绿。

关键文件：

- `.github/workflows/scripts/comment_ci_command.py`（模块 命令网关；类别 infra；类型 infrastructure；符号 `_validate_workflow_run_url`, `_request`, `create_workflow_dispatch`, `acknowledge_event`）：核心 handler：新增 `success_reaction` 注册表字段、`_validate_workflow_run_url`, `create_workflow_dispatch` 请求 run details，并支撑 `acknowledge/reply` 两条反馈入口。
- `.github/workflows/comment-ci-command.yml`（模块 CI 编排；类别 infra；类型 infrastructure）：工作流编排：新增 `acknowledge-command` 与 `reply-command` 两个独立 job，分别持有 Issues-only token 执行 +1 与链接回复，并传递 `success_reaction` / `workflow_run_url` 输出。
- `tests/ci/test/test_comment_ci_command.py`（模块 CI 测试；类别 test；类型 test-coverage；符号 `add_comment_reaction`, `create_issue_comment`, `test_preflight_writes_only_a_fixed_capability`, `test_preflight_writes_only_fixed_routing`）：验证主体：新增约 440 行测试，覆盖 `dispatch` 端点与 `run-detail` 校验、URL 输出传播、+1 与链接回复发布、token/job 隔离。
- `docs/ci/01-label.md`（模块 CI 文档；类别 docs；类型 documentation）：文档同步：更新命令网关行为描述，移除迁移期文案，补充 `/rerun-test` 的确认与链接回复契约。

关键符号：`create_workflow_dispatch`, `_validate_workflow_run_url`, `_request`, `acknowledge_event`, `reply_event`, `_write_capability`, `_write_routing`, `add_comment_reaction`, `create_issue_comment`

关键源码片段

`.github/workflows/scripts/comment_ci_command.py`

核心 handler：新增 `success_reaction` 注册表字段、`_validate_workflow_run_url`, `create_workflow_dispatch` 请求 run details，并支撑 `acknowledge/reply` 两条反馈入口。

命令注册表的扩展：每种命令声明自己的成功反馈方式

```
class CommandSpec(NamedTuple):
```

```
    policy_key: str
```

```
    capability: str # 决定走 issues 还是 actions 作用域
```

```
    handler: str
```

```
    audit_key: str
```

```
    audit_value: object
```

```
    success_reaction: str # 'none' 表示无反馈；'+1' 表示 dispatch 成功后打竖大拇指
```

只接受指向本仓库 Actions run 的完整 URL，拒绝外部或畸形地址

```
WORKFLOW_RUN_URL_PATTERN = re.compile(
```

```
    rf'https://github\.com/{re.escape(REPOSITORY)}/actions/runs/[1-9][0-9]*'
```

```
)
```

```
def _validate_workflow_run_url(value):
```

```
    if not isinstance(value, str) or WORKFLOW_RUN_URL_PATTERN.fullmatch(value) is None:
```

```

        raise CommentCommandError('workflow run URL is invalid')
    return value

# dispatch 核心：请求 GitHub 返回 run 详情，省去事后轮询 Actions 列表
def create_workflow_dispatch(self, workflow_file, ref, inputs):
    encoded_workflow = urllib.parse.quote(workflow_file, safe='')
    result = self._request(
        f'/repos/{REPOSITORY}/actions/workflows/{encoded_workflow}/dispatches',
        method='POST',
        payload={'ref': ref, 'inputs': inputs, 'return_run_details': True},
        expected_statuses=(200,), # 带详情时返回 200；旧的 204 表示已接受但无 run 信息
    )
    if not isinstance(result, dict):
        raise CommentCommandError('GitHub API returned invalid workflow dispatch details')
    run_id = _positive_int(result.get('workflow_run_id'), 'workflow run ID')
    html_url = _validate_workflow_run_url(result.get('html_url'))
    # 校验一致后只导出该 URL，后续由 reply job 发布链接回复
    return {'run_id': run_id, 'html_url': html_url}

```

评论区精华

PR 没有人工 review 评论线程：claude[bot] 只留下自动 review 提示，yueming-yuan 直接 APPROVED。设计论证集中在 PR body 的 Design Notes 与 Review Focus：可信命令注册表只把 `RunTestFile` 标记为需要反馈；Actions 作用域 job 重新授权调用者，以 `return_run_details: true` 调度固定 default-branch workflow，验证同仓库 run ID 与 URL 后只导出精确 `html_url`；两条反馈路径都依赖 dispatch 成功，且只持有 `Issues: write` App token。作者在第二个 commit 中特别说明，独立 Issues-scoped job 保证反馈失败不能重新 dispatch 测试。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 外部契约依赖：create_workflow_dispatch 依赖 GitHub API 在 return_run_details 时返回 workflow_run_id 与 html_url；若 API 改变返回结构或退回 204 空响应，dispatch 会 fail-closed，测试不触发也不会有错误反馈。
 2. 重复反馈：acknowledge-command 与 reply-command 手工重跑会在 PR 上叠加重复 +1 或重复链接评论，工作流没有幂等保护，body 也已明确不自动重试含糊响应。
 3. 可见性风险：反馈失败时已接受的测试仍会运行，用户看不到确认可能重复发送 /rerun-test，导致重复 dispatch。
 4. URL 校验边界：_validate_workflow_run_url 校验的是 URL 形状与本仓库前缀；若 API 返回同仓库但非本次 run 的 URL，仍可能被展示，依赖 run_id 的一致性比对兜底。
 5. 签名兼容：_request 改签名后所有调用点必须同步，漏改会直接 TypeError；当前测试覆盖主要端点。
- 影响：

1. 开发者：发 `/rerun-test <test-file>` 后会立即收到 +1 和直达 Actions run 的链接，调试反馈链路明显缩短；标签命令与 `/rerun-failed-ci` 路由不变。
2. 系统：CI 命令网关新增两个轻量 job 和一次带 run-details 的 dispatch 调用，Actions 开销可忽略；反馈 job 不执行 PR 代码，不引入恶意代码执行面。
3. 团队：命令网关形成统一的“接受 + 反馈”模式，后续新增命令只需在注册表声明 `success_reaction` 即可复用。
4. 范围：仅影响 CI/PR 自动化，不触及训练、rollout、模型等核心路径。 - 风险标记：依赖 GitHub dispatch API 返回结构，反馈失败不重试不重调度，手动重跑可产生重复回复，URL 校验依赖外部契约

关联脉络

- PR #2496 feat(ci): add authorized comment-to-label gateway: 本 PR 是同一命令网关的直接后继，修改完全相同的四个文件（`comment_ci_command.py`、`comment-ci-command.yml`、`docs/ci/01-label.md`、`tests/ci/test/test_comment_ci_command.py`），在授权网关之上增加确认反馈。
- PR #2675 [CI] rebuild a PR's docker image only when its build inputs change: 同期 CI 基础设施演进，反映团队在 PR 自动化质量与开发者体验上的持续投入；虽文件不重叠，但同属 CI 工具链优化。