

Publishing a Custom Action



Kamran Ayub

Helping you prepare for the GitHub Actions exam

www.linkedin.com/in/kamranayub

Distributing Private or Public Actions

A **private action** is stored in a **private repository**

A **public action** is stored in a **public repository**

Publishing to the GitHub Marketplace is an **optional step**



Private Actions



MUST be in a private repository

MAY be consumed by other private repos

- Only by private repos owned by the personal or organizational account
- Sharing can be enabled in the action repository's settings

MAY NOT be consumed by public workflows

- Public or private repositories not owned by the user or organization will not have access



Public Actions



MUST be in a public repository

**MAY be consumed by private or public
repository workflows across GitHub**



Why publish to the GitHub Marketplace?

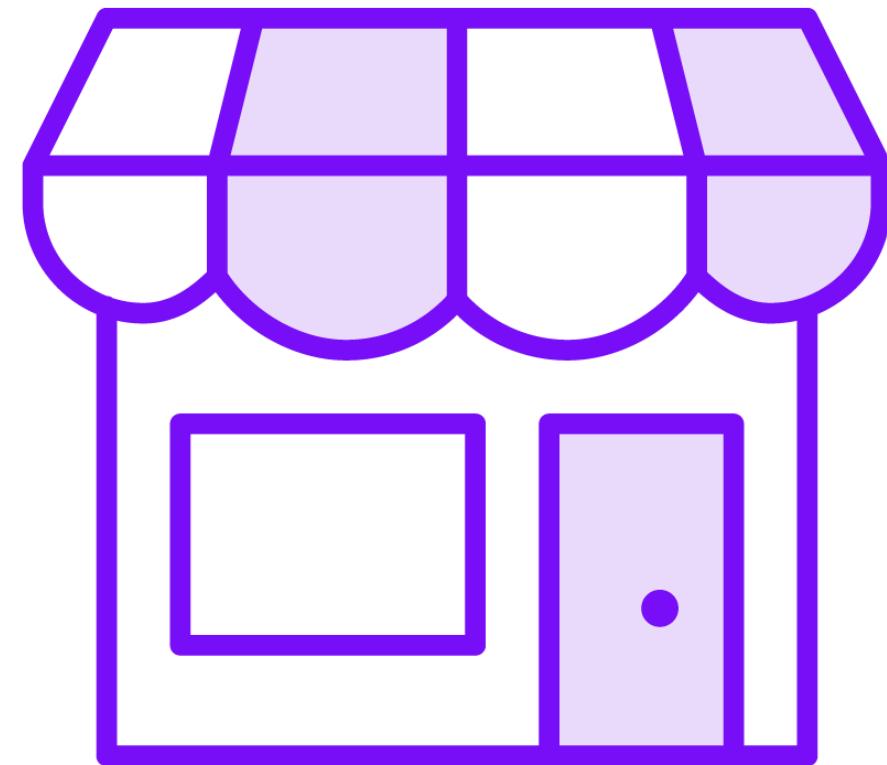


Why publish to the GitHub Marketplace?

Discoverability.



GitHub Marketplace Actions



MUST have a unique name

- It cannot conflict with an existing name in the Marketplace or Marketplace category
- You may publish an action that matches the name of your user or org

MUST accept the GitHub Marketplace Developer Agreement

- This must be accepted by the account or organizational owner



One Action Per Repo Limitation



MUST have a single action metadata at the root

- GitHub Marketplace will use the one in the main branch

MAY use separate branches for multiple actions in a single repo if not publishing

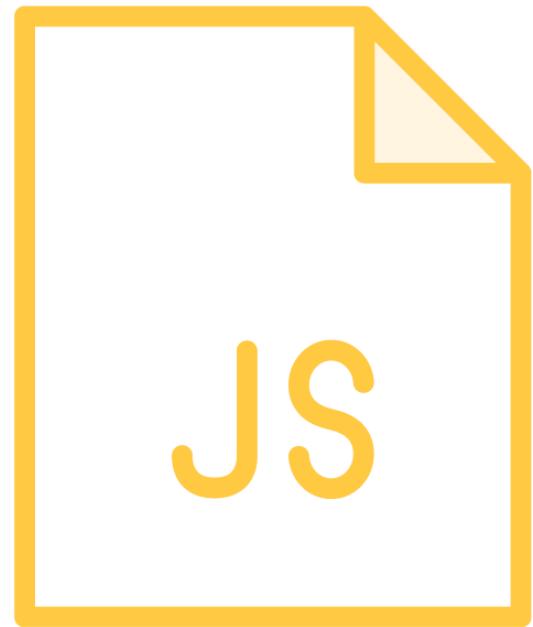
- uses: `owner/action-repo@BRANCH`
- This is harder to maintain

MAY use a top-level composite action to compose multiple actions

- If you are just trying to call multiple custom actions in the same workflow



Handling Dependencies in JavaScript Actions



Installing dependencies can be non-deterministic for traditional Node.js projects

GitHub Actions need to be deterministic to avoid workflow failures

MUST have dependencies stored in Git when consuming the action from a workflow

- You may commit node_modules but this can bloat your Git repository
- Instead, you can use `@vercel/ncc` to compile your code to a single file in the dist directory



Compiling a JavaScript Action Using ncc

This will output a single `index.js` file into a `dist` directory that the action can run

`package.json`

```
"devDependencies": {  
  "@vercel/ncc": "latest",  
},  
"scripts": {  
  "build": "npx ncc build src/index.js -o dist"  
}
```

`action.yml`

```
runs:  
  using: node20  
  main: dist/index.js
```

```
> npm install --save-dev @vercel/ncc  
> npm run build
```

References

Sharing actions and workflows from your private repository

<https://docs.github.com/en/actions/sharing-automations/sharing-actions-and-workflows-from-your-private-repository>

Sharing actions and workflows with your organization

<https://docs.github.com/en/actions/sharing-automations/sharing-actions-and-workflows-from-your-private-repository>

Publishing actions in GitHub Marketplace

<https://docs.github.com/en/actions/sharing-automations/creating-actions/publishing-actions-in-github-marketplace>

Build and Tag Action for Publishing JavaScript Actions

<https://github.com/JasonEtco/build-and-tag-action>



**Sharing an action means
you are now an open source
maintainer.**



Recommendations for Sharing Your Action



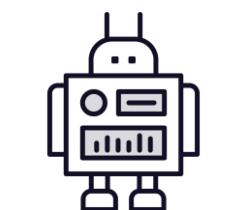
Add integration test workflows and unit tests



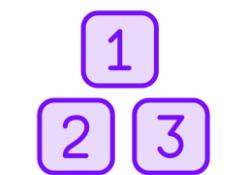
Add a README that explains how (and when) to use the action



Explain how people should report issues and contribute



Automate releases with GitHub tagging and dependency updates



Use semantic versioning to avoid breaking workflows (e.g. @v2)



**You may be tempted to not
follow any of these
practices. Don't give in!**



Recommendations for Sharing Your Action



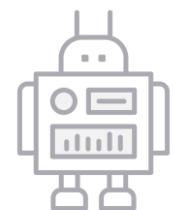
Add integration test workflows and unit tests



Add a README that explains how (and when) to use the action



Explain how people should report issues and contribute



Automate releases with GitHub tagging and dependency updates



Use semantic versioning to avoid breaking workflows (e.g. @v2)



Recommendations for Sharing Your Action



Add integration test workflows and unit tests



Add a README that explains how (and when) to use the action



Explain how people should report issues and contribute



Automate releases with GitHub tagging and dependency updates



Use semantic versioning to avoid breaking workflows (e.g. @v2)



Demo: Current README

```
1 # pluralsight-authoring-github-actions-demo-action
2
3 A demo containing multiple custom GitHub Actions:
4
5 - [action.yml](./action.yml) -- A Composite GitHub action that gets the Slack ID for a GitHub actor and sends them a Slack message through the slackapi/slack-github-action action.
6 - [get-employee-js-action](./get-employee-js-action/action.yml) -- A JavaScript action that looks up the Slack User ID for the GitHub actor who triggered the workflow.
7
```



Demo: JS Action Template README

Hello, World! JavaScript Action

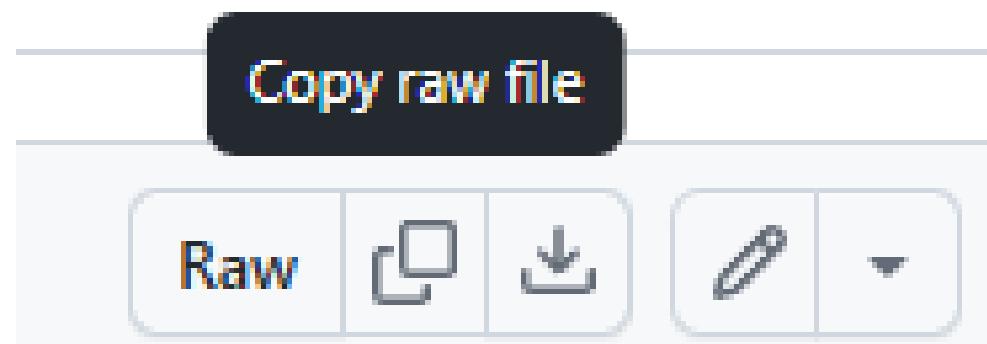
 Lint Codebase passing  Continuous Integration passing

This action prints `Hello, World!` or `Hello, <who-to-greet>!` to the log. To learn how this action was built, see [Creating a JavaScript action](#).

Create Your Own Action

To create your own action, you can use this repository as a template! Just follow the below instructions:

1. Click the [Use this template](#) button at the top of the repository
2. Select [Create a new repository](#)
3. Select an owner and name for your new repository
4. Click [Create repository](#)
5. Clone your new repository



Demo: New README

Explain what the action does and when to use it, with an example workflow

README.md

```
# Notify Slack User Demo Action (Pluralsight)
```

```
![Build and Test](https://github.com/kamranayub/pluralsight-authoring-github-actions-demo-action/actions/workflows/test.yml/badge.svg)
```

This action sends a Slack message to a GitHub user by looking them up in your Slack workspace by their public email address. This is a demo action for the [Authoring and Maintaining GitHub Actions Pluralsight course](https://github.com/kamranayub/pluralsight-course-authoring-maintaining-github-actions).



Demo: New README (Cont.)

Explain what the action does and when to use it, with an example workflow

README.md

```
> [ !CAUTION]  
>  
> DO NOT USE THIS ACTION. It is a working demo for the course and  
> is not intended for public consumption. I will not be maintaining  
> it beyond the scope course updates.
```



Demo: New README (Cont.)

Explain what the action does and when to use it, with an example workflow

README.md

```
## Usage

Here's an example of how to use this action in a workflow file:

```yaml
name: Example Workflow

on:
 pull_request:
 branches:
 - main

jobs:
 build-and-test:
 name: Example build and test workflow
 runs-on: ubuntu-latest

 steps:
 - name: Checkout
 uses: actions/checkout@v4

 #
 # Perform your project build steps here
 #

 # TODO: Replace with actual SHA/version for published action
 # Change @main to a specific commit SHA or version tag, e.g.:
 # kamranayub/pluralsight-authoring-github-actions-demo-action@e76147da8e5c81eaf017dede5645551d4b94427b
 # kamranayub/pluralsight-authoring-github-actions-demo-action@v1.0.0
 - id: "notify-slack"
 name: "Send Slack notification to user"
 uses: kamranayub/pluralsight-authoring-github-actions-demo-action@main
 with:
 github-username: ${{ github.actor }}
 github-token: ${{ secrets.GITHUB_TOKEN }}
 slack-token: ${{ secrets.SLACK_TOKEN }}
 slack-bot-token: ${{ secrets.SLACK_BOT_TOKEN }}
```

```

For example workflow runs, check out the [Actions tab](<https://github.com/kamranayub/pluralsight-authoring-github-actions-demo-action/actions>)!
:rocket:



Demo: New README (Cont.)

Explain what the action does and when to use it, with an example workflow

README.md

Prerequisites

- You need to set up a Slack app. [Follow the instructions](<https://github.com/slackapi/slack-github-action?tab=readme-overview#technique-2-slack-app>) for "Technique 2: Slack App" for the Slack Send action to set up your `SLACK_BOT_TOKEN` secret.
- Add the two Slack secrets to your repository, `SLACK_TOKEN` and `SLACK_BOT_TOKEN`
 - These can be the same tokens but the `SLACK_BOT_TOKEN` requires the `chat:write` OAuth scope.
- Ensure the GitHub user(s) you want to notify have their public profile email set to whatever email they use for your Slack workspace. If their email is private, or if the email doesn't match a user in Slack, the action will not work.



Demo: New README (Cont.)

Explain what the action does and when to use it, with an example workflow

README.md

```
## Inputs

Input Description	Default	Required
`github-token`	true	The GitHub token to authenticate with GitHub API
`github-username`	true	The GitHub username of the employee to look up
`slack-token`	true	The organizational Slack access token to look up users
`slack-bot-token`	true	The organizational Slack access token to send messages

## Outputs

Output	Description
`time`	The time the message was sent
```

Copilot Prompt

The screenshot shows the GitHub Copilot interface with the following details:

- User Profile:** kamranayub
- Description:** Take the inputs and Outputs from the action Yaml file and format them using the same format that I've highlighted the code in markdown.
- Action YML:** action.yml
- GitHub Copilot:** Used 2 references
- Inputs:**

| Input | Default | Description |
|-------------------|---------|--|
| `github-token` | | The GitHub token to authenticate with GitHub API |
| `github-username` | | The GitHub username of the employee to look up |
| `slack-token` | | The organizational Slack access token to look up users |
| `slack-bot-token` | | The organizational Slack access token to send messages |
- Outputs:**

| Output | Description |
|--------|-------------------------------|
| `time` | The time the message was sent |
- Search attachments:** Search attachments
- Selection:** Selection
- Symbol...**
- Terminal Selection**
- VS Code API**
- README.md**
- test.yml** .github\workflows
- action.yml**
- README.md get-employee-docker-action\test**
- Recently Opened:** recently opened
- User Profile:** kamranayub
- Description:** Thanks. Can you also add a required column to the table with the value
- GitHub Copilot:** Used 1 reference
- Inputs:**

| Input | Default | Required | Description |
|-------------------|---------|----------|--|
| `github-token` | | true | The GitHub token to authenticate with GitHub API |
| `github-username` | | true | The GitHub username of the employee to look up |
| `slack-token` | | true | The organizational Slack access token to look up users |
| `slack-bot-token` | | true | The organizational Slack access token to send messages |
- Outputs:**

| Output | Description |
|--------|-------------------------------|
| `time` | The time the message was sent |

Demo: New README (Cont.)

Explain what the action does and when to use it, with an example workflow

README.md

```
> [ !NOTE ]
>
> The `slack-token` and `slack-bot-token` can be the same token, but the `slack-
bot-token` must have the necessary `chat:write` OAuth scope.
>
> Please see the [slack-send](https://github.com/slackapi/slack-github-action?tab=readme-ov-file#technique-2-slack-app) action for details on the bot
token.
```



Demo: New README (Cont.)

Explain what the action does and when to use it, with an example workflow

README.md

How to Use the `time` Output

The `time` output from the action provides the timestamp of when the Slack message was sent. This can be useful for logging or further processing in your workflow.

Here is an example of how to use the `time` output in your GitHub Actions workflow:

```
```yaml
- id: "notify-slack"
 name: "Send Slack notification to user"
 uses: kamranayub/pluralsight-authoring-github-
actions-demo-action@main
 with:
 github-username: ${{ github.actor }}
 github-token: ${{ secrets.GITHUB_TOKEN }}
 slack-token: ${{ secrets.SLACK_TOKEN }}
 slack-bot-token: ${{ secrets.SLACK_BOT_TOKEN }}

 - name: Print Slack Message Time
 run: echo "Message sent at: ${{ steps.notify-
slack.outputs.time }}"
```

```

Copilot Prompt

Can you write a little bit of markdown documentation explaining, here's how to use the time output based on the existing usage example in the README file?



References

About READMEs

<https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>

Awesome READMEs

<https://github.com/matiassingers/awesome-readme>



Recommendations for Sharing Your Action



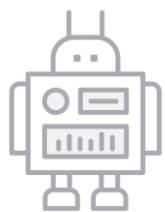
Add integration test workflows and unit tests



Add a README that explains how (and when) to use the action



Explain how people should report issues and contribute



Automate releases with GitHub tagging and dependency updates



Use semantic versioning to avoid breaking workflows (e.g. @v2)



Demo: Referencing GitHub Community Health Files

Code of conduct MIT license Security

.github Public Watch 23

main 3 Branches 0 Tags Go to file Add file Code

| leahwicz | Merge pull request #23 from konradpabjan/patch-1 | f7b5c7c · 4 years ago | 17 Commits |
|--------------------|--|-----------------------|------------|
| .github | Fix busted link in issue template | 4 years ago | |
| CODEOWNERS | Fix CODEOWNERS team | 4 years ago | |
| CODE_OF_CONDUCT.md | standard code of conduct | 5 years ago | |
| SECURITY.md | Update SECURITY.md | 5 years ago | |



Demo: Customizing Community Health Files

Reporting security issues

.github/SECURITY.md

If you discover a security issue in this repo, please email me through my GitHub profile ([kamranayub](#)).

Thanks for helping make GitHub Actions safe for everyone.



Demo: Customizing Community Health Files

Code of Conduct

```
.github/CODE_OF_CONDUCT.  
md
```

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting me through my GitHub profile (kamranayub).



Demo: Customizing Community Health Files

Contribution guidelines

.github/CONTRIBUTING.md

```
# Contribution guidelines

## Reporting bugs or issues

If there's an issue that prevents the course demo from working or is an issue with the learning material, please open an issue and report it so I can address it for the next course update.

Please do not open issues that are about enhancing or changing the action, since this is just for demo purposes and not for production consumption.

## Environment setup

You can use the provided DevContainer with GitHub Codespaces which should be pre-configured with all these tools, but there are some additional third-party services and tokens you'll need outlined below.

### Tools

- To work on the JavaScript action, you will need:
  - [Node 20](https://nodejs.org) or above
  - I use [fnm](https://github.com/Schniz/fnm) for Windows to manage node versions
  - The exact version is stored in the `nvmrc` file for the JavaScript action
- To work on the Docker action, you will need:
  - [Docker Desktop 27+](https://www.docker.com/products/docker-desktop/)
  - [.NET 8+](https://get.dot.net)
  - On Windows, you also need [WSL2 setup](https://learn.microsoft.com/en-us/windows/wsl/install) (Ubuntu 22+)

### Third-Party Services

- A [public GitHub profile](https://docs.github.com/en/account-and-profile/setting-up-and-managing-your-github-profile/customizing-your-profile/personalizing-your-profile) with a public email set
- A Slack app connected to a test workspace ([follow the steps in the Slack Send action](https://github.com/slackapi/slack-github-action?tab=readme-ov-file#technique-2-slack-app))

### Environment variables

- `GITHUB_TOKEN` -- A personal access token (PAT) that has access to your user profile (public permissions are fine)
- `SLACK_TOKEN` -- A Slack app OAuth bot token (no special permissions required) for your Slack workspace
- `SLACK_BOT_TOKEN` -- A Slack app OAuth bot token (with `chat:write` OAuth scope) for your Slack workspace

## Branches

- `main` - The "published" version of the action
- `m3/demo` - The Module 3 final demo material
- `m4/demo` - The Module 4 final demo material
- `m5/demo` - The Module 5 final demo material
```



Demo: Customizing Community Health Files

LICENSE

LICENSE

Copyright 2024 Kamran Ayub



References

Building communities

<https://docs.github.com/en/communities>

Creating a default community health file

<https://docs.github.com/en/communities/setting-up-your-project-for-healthy-contributions/creating-a-default-community-health-file>



Recommendations for Sharing Your Action



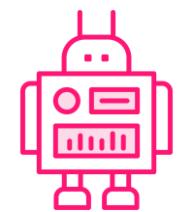
Add integration test workflows and unit tests



Add a README that explains how (and when) to use the action



Explain how people should report issues and contribute



Automate releases with GitHub tagging and dependency updates



Use semantic versioning to avoid breaking workflows (e.g. @v2)



Compiling a JavaScript Action Using ncc

This will output a single `index.js` file into a `dist` directory that the action can run

`package.json`

```
"devDependencies": {  
  "@vercel/ncc": "<version>,"  
},  
"scripts": {  
  "build": "npx ncc build main.js -o dist"  
}
```

`action.yml`

```
runs:  
  using: node20  
  main: dist/index.js
```

```
> npm install @vercel/ncc --save-dev  
> npm run build
```

Demo: How ncc works

get-employee-js-action > **JS** main.js > ...
You, 5 days ago | 2 authors (You and one other)
1 // @ts-check Kamran Ayub, last month • initial demo
2 const core = require("@actions/core");
3 const github = require("@actions/github");
4 const { WebClient } = require("@slack/web-api");
5

get-employee-js-action
└── dist
 └── **JS** index.js
node_modules
└── .bin
└── @actions
└── @fastify
└── @octokit
└── @slack
└── @types
└── @vercel
└── asynckit
└── axios

```
get-employee-js-action > dist > JS index.js
1 //***** ( () => { // webpackBootstrap
2 //***** var __webpack_modules__ = ( {
3
4 //*** 7351:
5 /**/ (function(__unused_webpack_module, exports, __nccwpck_require__) {
6
7 "use strict";
8
9 var __createBinding = (this && this.__createBinding) || (Object.create ? (function(o, m, k, k2) {
10 if (k2 === undefined) k2 = k;
11 Object.defineProperty(o, k2, { enumerable: true, get: function() { return m[k]; } });
12 }) : (function(o, m, k, k2) {
13 if (k2 === undefined) k2 = k;
14 o[k2] = m[k];
15 }));
16 var __setModuleDefault = (this && this.__setModuleDefault) || (Object.create ? (function(o, v) {
17 Object.defineProperty(o, "default", { enumerable: true, value: v });
18 }) : function(o, v) {
19 o["default"] = v;
20 }));

```



Compiling a JavaScript Action Using ncc

Commit the dist directory

.gitignore

```
# Remove dist directory
```

```
On branch m5/versioning
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   ./.gitignore
    new file:   dist/index.js
    modified:   package-lock.json
    modified:   package.json
```

```
> git status
> git add -A
> git commit -m "add ncc"
> git push --set-upstream
origin m5/demo-building
```

[Open PR]

[Open workflow status]

|  Test Get Employee Action (JavaScript)

Demo: Creating a Release Version Tag

The screenshot illustrates the process of creating a new release tag on GitHub.

Create a new release interface:

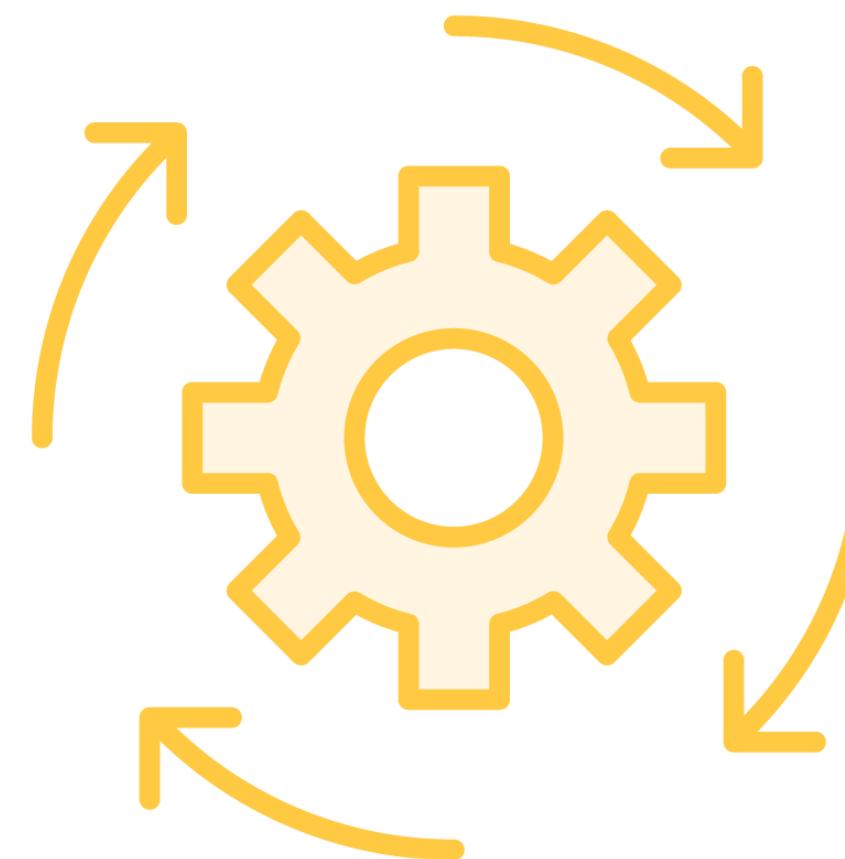
- Header:** Choose a tag (dropdown), Target: m5/versioning (dropdown), Generate release notes (button).
- Text input:** v0.1.0
- Toolbar:** Write (selected), Preview, and various rich text editing icons.
- Content area:** FOR DEMO PURPOSES ONLY
- Bottom controls:** Create a new release (button).

Resulting release page:

- Header:** v0.1.0 (Pre-release)
- Author:** kamranayub released this now (with profile icon)
- Assets:** 2 items:
 - Source code (zip) uploaded 8 minutes ago
 - Source code (tar.gz) uploaded 8 minutes ago
- Release notes:** FOR DEMO PURPOSES ONLY
- File upload:** Paste, drop, or click to add files
- Binary attachment:** Attach binaries by dropping them here or selecting them.
- Checkboxes:** Set as a pre-release (checked, This release will be labeled as non-production ready)
- Buttons:** Publish release (green button), Save draft



Tips for Building JavaScript Actions



Use a workflow to verify the dist files were updated

- The [github/javascript-action](#) template has an example (check-dist)

You could try the [JasonEtco/build-and-tag-action](#)

- This avoids committing dist to main and will automatically build and tag new releases
- This is community-maintained and may not always be compatible with the latest Actions infrastructure



References

Check Dist Workflow

<https://github.com/actions/javascript-action/blob/main/.github/workflows/check-dist.yml>

Build and Tag Action (Community-Owned)

<https://github.com/JasonEtco/build-and-tag-action>



Demo: Enabling Dependency Updates

General

Access

Collaborators

Code and automation

- Branches
- Tags
- Rules
- Actions
- Webhooks
- Environments
- Codespaces
- Pages

Security

- Code security and analysis** (selected)
- Deploy keys
- Secrets and variables

Integrations

- GitHub Apps
- Email notifications
- Autolink references

Code security and analysis

Security and analysis features help keep your repository secure and updated. By enabling these features, you're granting us permission to perform read-only analysis on your repository.

Dependency graph
Understand your dependencies. [Enable](#)

Dependabot
Keep your dependencies secure and up-to-date. [Learn more about Dependabot](#).

Dependabot alerts
Receive alerts for vulnerabilities that affect your dependencies and manually generate Dependabot pull requests to resolve these vulnerabilities. [Configure alert notifications](#). [Enable](#)

Dependabot security updates
Enabling this option will result in Dependabot automatically attempting to open pull requests to resolve every open Dependabot alert with an available patch. [Enable](#)

Grouped security updates
Groups all available updates that resolve a Dependabot alert into one pull request (per package manager and directory of requirement manifests). This option may be overridden by group rules specified in dependabot.yml - [learn more here](#) [Enable](#)

Dependabot version updates
Allow Dependabot to open pull requests automatically to keep your dependencies up-to-date when new versions are available. [Learn more about configuring a dependabot.yml file](#). [Enable](#)

Dependabot on Actions runners
Run Dependabot security and version updates on Actions runners. [Enable](#)

Dependabot on self-hosted runners
Run Dependabot security and version updates on self-hosted Actions runners. [Enable](#)



Recommendations for Sharing Your Action



Add integration test workflows and unit tests



Add a README that explains how (and when) to use the action



Explain how people should report issues and contribute



Automate releases with GitHub tagging and dependency updates



Use semantic versioning to avoid breaking workflows (e.g. @v2)



Demo: Enabling Sharing for Private Repos

Access

Control how this repository is used by GitHub Actions workflows in other repositories. [Learn more about allowing other repositories to access to Actions components in this repository.](#)

Not accessible

Workflows in other repositories cannot access this repository.

Accessible from repositories owned by the user 'kamranayub'

Workflows in other repositories that are owned by the user 'kamranayub' can access the actions and reusable workflows in this repository.
Access is allowed only from private repositories.

Save



Demo: Create a Consumer Workflow

Try consuming private action

.github/workflows/demo.yml

```
name: Demo Consumer

on:
  workflow_dispatch:

jobs:
  test-demo-action:
    name: Test Pluralsight Demo Action
    runs-on: ubuntu-latest

steps:
  - id: test-send-action
    uses: kamranayub/pluralsight-authoring-github-actions-demo-action@v0.1.0
    with:
      github-username: ${{ github.actor }}
      github-token: ${{ secrets.GITHUB_TOKEN }}
      slack-token: ${{ secrets.SLACK_TOKEN }}
      slack-bot-token: ${{ secrets.SLACK_BOT_TOKEN }}

  - name: Print Slack Message Time
    run: echo "${{ steps.test-send-action.outputs.time }}"
```



Demo: Failing Due to Local Action Path

Error: Can't find 'action.yml', 'action.yaml' or 'Dockerfile' under '/home/runner/work/pluralsight-demo-action-consumer/pluralsight-demo-action-consumer/get-employee-js-action'. Did you forget to run actions/checkout before running your local action?



Demo: Create a Consumer Workflow

Fix running local actions when consuming a composite action

action.yml

```
- name: Set reference to current action version
  id: set-action-version
  run: echo "ref=$ACTION_REF" >> $GITHUB_OUTPUT
  shell: bash
  env:
    ACTION_REF: ${{ github.action_ref }}

# See: https://github.com/orgs/community/discussions/67904
- name: Setup workaround for local actions
  uses: actions/checkout@v4
  with:
    repository: kamranayub/pluralsight-authoring-github-actions-demo-action
    ref: ${{ steps.set-action-version.outputs.ref }}
    token: ${{ inputs.github-token }}
    path: .local-action-cache
    sparse-checkout:
      - get-employee-js-action

- id: get-employee-js
  uses: './.local-action-cache/get-employee-js-action'
```



Demo: Create a Consumer Workflow

Try consuming private action

.github/workflows/demo.yml

```
name: Demo Consumer

on:
  workflow_dispatch:

jobs:
  test-demo-action:
    name: Test Pluralsight Demo Action
    runs-on: ubuntu-latest

    steps:
      - id: test-send-action
        uses: kamranayub/pluralsight-authoring-github-actions-demo-action@v0.2.0
        with:
          github-username: ${{ github.actor }}
          github-token: ${{ secrets.GITHUB_TOKEN }}
          slack-token: ${{ secrets.SLACK_TOKEN }}
          slack-bot-token: ${{ secrets.SLACK_BOT_TOKEN }}

      - name: Print Slack Message Time
        run: echo "${{ steps.test-send-action.outputs.time }}"
```



Demo: Add PRIVATE_GITHUB_TOKEN secret

Repository secrets

New repository secret

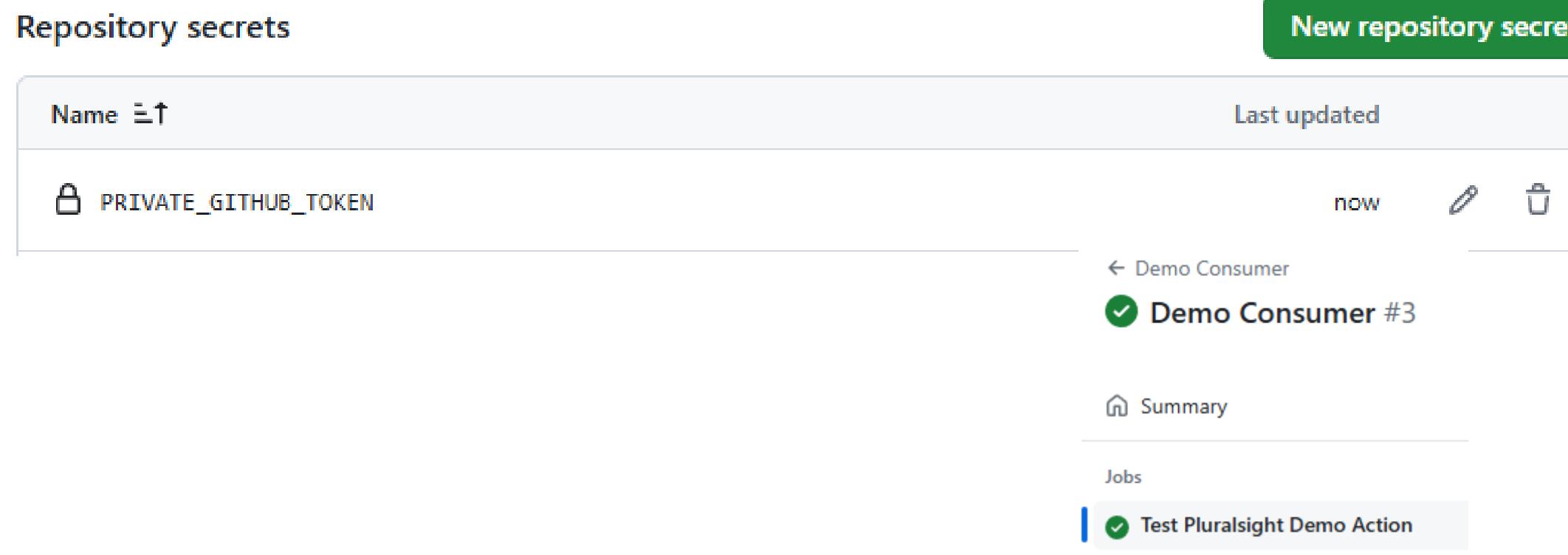
| Name ↗↑ | Last updated |
|--|---|
|  PRIVATE_GITHUB_TOKEN | now   |

← Demo Consumer
✓ Demo Consumer #3

 Summary

Jobs

✓ Test Pluralsight Demo Action



Demo: Successfully consume action

← Demo Consumer

✓ Demo Consumer #3

Summary

Jobs

| ✓ Test Pluralsight Demo Action



Pluralsight Demo GitHub Actions APP 12:37 AM
GitHub build result: success

12:38 GitHub build result: success



**Consuming your action
privately at first is a good
way to fully test that it
works.**



References

Sharing actions and workflows from your private repository

<https://docs.github.com/en/actions/sharing-automations/sharing-actions-and-workflows-from-your-private-repository>

Sharing actions and workflows with your organization

<https://docs.github.com/en/actions/sharing-automations/sharing-actions-and-workflows-from-your-private-repository>

Composite action failure discussion

<https://github.com/orgs/community/discussions/67904>



Demo: Publishing to the GitHub Marketplace

A screenshot of a GitHub repository page. At the top, there are navigation links for 'v0.2.0' (with a dropdown arrow), '5 Branches' (with a dropdown arrow), '2 Tags' (with a dropdown arrow), a search bar with placeholder 'Go to file' and a 't' icon, and a green 'Code' button. Below this is a table of recent commits:

| Author | Commit Message | Time Ago |
|----------------------------|---|-----------------------|
| kamranayub | expand env var ✓ | 9dca828 · 7 hours ago |
| .github | add community health files | 16 hours ago |
| .vscode | M4: Troubleshooting for composite action (#8) | last week |
| get-employee-docker-action | add docker test script | yesterday |
| get-employee-js-action | fix main | 8 hours ago |
| .gitignore | add ncc | 10 hours ago |
| LICENSE | add community health files | 16 hours ago |
| README.md | badge | 9 hours ago |
| action.yml | expand env var | 7 hours ago |

At the bottom of the repository view, there are links for 'README', 'Code of conduct', 'License', and 'Security'. There is also a three-dot menu icon.

A screenshot of the 'Notify Slack User Demo Action (Pluralsight)' page on the GitHub Marketplace. The title is 'Notify Slack User Demo Action (Pluralsight)'. Below it is a 'Test Actions' button with a green 'passing' status indicator. A descriptive text block states: 'This action sends a Slack message to a GitHub user by looking them up in your Slack workspace by their public email address. This is a demo action for the [Authoring and Maintaining GitHub Actions Pluralsight course](#)'. A red-bordered 'Caution' section contains the text: 'DO NOT USE THIS ACTION. It is a working demo for the course and is not intended for public consumption. I will not be maintaining it beyond the scope course updates.' In the bottom right corner of the page, there is a circular icon with a white play button symbol.

Setting Up Action Metadata

Use a good description

action.yml

```
name: 'Send Build Status Notification to Slack User (DEMO)'  
description: "Reports the build status to the GitHub user who triggered the  
workflow via Slack. This is a demo action, DO NOT USE."
```



Setting Up Action Metadata

Set optional branding attributes

action.yml

```
branding:  
  color: red  
  icon: send
```



Demo: Make Repository Public

Danger Zone

[Change repository visibility](#)

This repository is currently public.

[Change visibility](#)



Demo: Tag and Publish to Marketplace

You can publish this Action to the GitHub Marketplace

[Draft a release](#) [X](#)

Release Action

Publish this release to the GitHub Marketplace [?](#)
You must accept the [GitHub Marketplace Developer Agreement](#) before publishing an Action.

[v0.3.0](#) [Target: m5/publish](#) [Previous tag: auto](#) [Generate release notes](#)

Excellent! This tag will be created from the target when you publish this release.

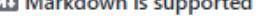
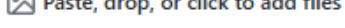
Release title

[Write](#) [Preview](#)

DEMO PUBLIC RELEASE, DO NOT USE.

By clicking "Accept Terms" below, you agree to the GitHub Marketplace Developer Agreement v2.4 on behalf of  kamranayub

[Accept Terms](#)

 Markdown is supported  Paste, drop, or click to add files

↓ Attach binaries by dropping them here or selecting them.

Set as a pre-release
This release will be labeled as non-production ready

[Publish release](#) [Save draft](#)



Demo: Publish Checklist

Release Action

Publish this Action to the GitHub Marketplace

Your Action will be discoverable in the Marketplace and available in GitHub search.

 **action.yml** 

✓ Everything looks good! You have all the required information.

| | |
|---------------|--|
| ✓ Name | Send Build Status Notification to Slack User (D...) |
| ✓ Description | Reports the build status to the GitHub user who triggered the workflow via Slack. This is a demo ... |
| ✓ Icon | send |
| ✓ Color | red |

 **README** 

✓ A README exists.

Primary Category  **Continuous integration**

Another Category — optional  **Reporting**



Demo: Publish Release

v0.4.0 Target: main Previous tag: auto Generate release notes

Excellent! This tag will be created from the target when you publish this release.

v0.4.0

Write Preview H B I E <> C | E E E E | @ ↵ ↶ ↷ ↸ ↹ ↺

DEMO PUBLIC RELEASE FOR MARKETPLACE. DO NOT USE. Not intended for public consumption!

Markdown is supported Paste, drop, or click to add files

↓ Attach binaries by dropping them here or selecting them.

Set as a pre-release
This release will be labeled as non-production ready

Set as the latest release
This release will be labeled as the latest for this repository.

Publish release Save draft



Demo: It's Live!

The screenshot shows the GitHub Marketplace interface. At the top, there is a navigation bar with icons for search, filters, and user profile. Below the navigation, the URL is displayed as [Marketplace / Actions / Send Build Status Notification to Slack User \(DEMO\)](#). A 'Delist' button is also visible.

The main content area displays a GitHub Action card for 'Send Build Status Notification to Slack User (DEMO)'. The card includes a red circular icon with a white paper airplane icon, the action name, its version (v0.4.0), and a 'Latest version' button. A dropdown menu next to the version button shows 'Use latest version'.

Below the card, the title 'Notify Slack User Demo Action (Pluralsight)' is shown. Underneath, there is a 'Test Actions' section indicating 'passing'. A note states: 'This action sends a Slack message to a GitHub user by looking them up in your Slack workspace by their public email address. This is a demo action for the [Authoring and Maintaining GitHub Actions Pluralsight course](#)'. A 'Caution' section warns: 'DO NOT USE THIS ACTION. It is a working demo for the course and is not intended for public consumption. I will not be maintaining it beyond the scope course updates.'

The 'Prerequisites' section lists: 'You need to set up a Slack app. [Follow the instructions](#) for "Technique 2: Slack App" for the Slack Send action to set up your `SLACK_BOT_TOKEN` secret.'

On the right side of the page, there are sections for 'Stars' (0), 'Contributors' (1), 'Categories' (Continuous Integration, Reporting), and 'Links' (repository link, 0 open issues, 1 pull request, report abuse).



Demo: Test Demo Consumer

kamranayub / pluralsight-demo-action-consumer

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

pluralsight-demo-action-consumer/.github/workflows/demo.yml in main

Edit Preview

```
1 name: Demo Consumer
2
3 on:
4   workflow_dispatch:
5
6 jobs:
7   test-demo-action:
8     name: Test Pluralsight Demo Action
9     runs-on: ubuntu-latest
10
11 steps:
12   - id: test-send-action
13     uses: kamranayub/pluralsight-authoring-github-actions-demo-action@v0.4.0
14     with:
15       github-username: ${{ github.actor }}
16       github-token: ${{ secrets.PRIVATE_GITHUB_TOKEN }}
17       slack-token: ${{ secrets.SLACK_TOKEN }}
18       slack-bot-token: ${{ secrets.SLACK_BOT_TOKEN }}
19
```

Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page.

Use `Control + Space` to trigger autocomplete in most situations.

← Demo Consumer

Demo Consumer #4

Summary

Jobs

Test Pluralsight Demo Action

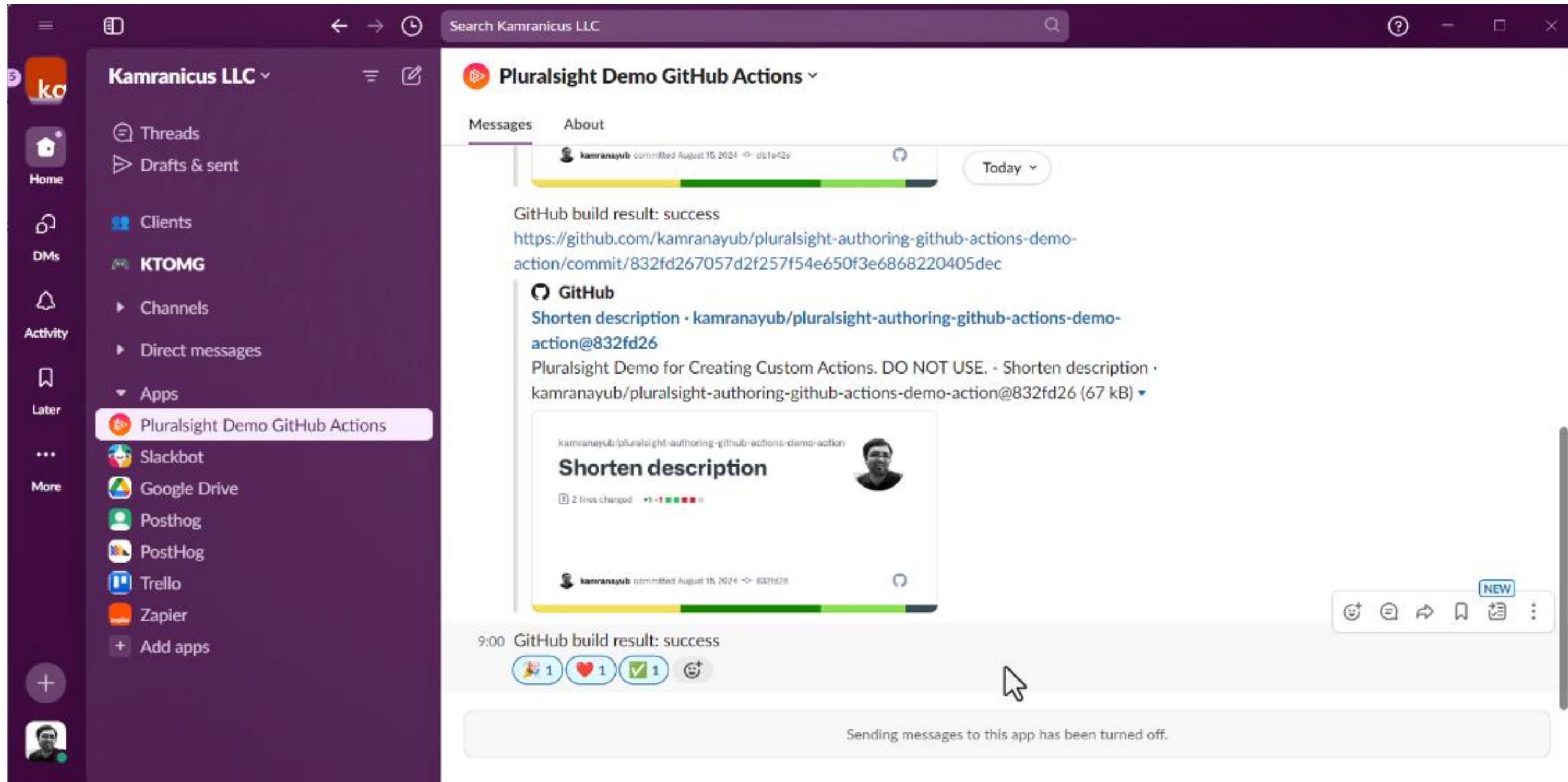
Run details

Usage

Workflow file



Demo: Slack Messages



Demo: Remove Action from Marketplace

Releases Tags

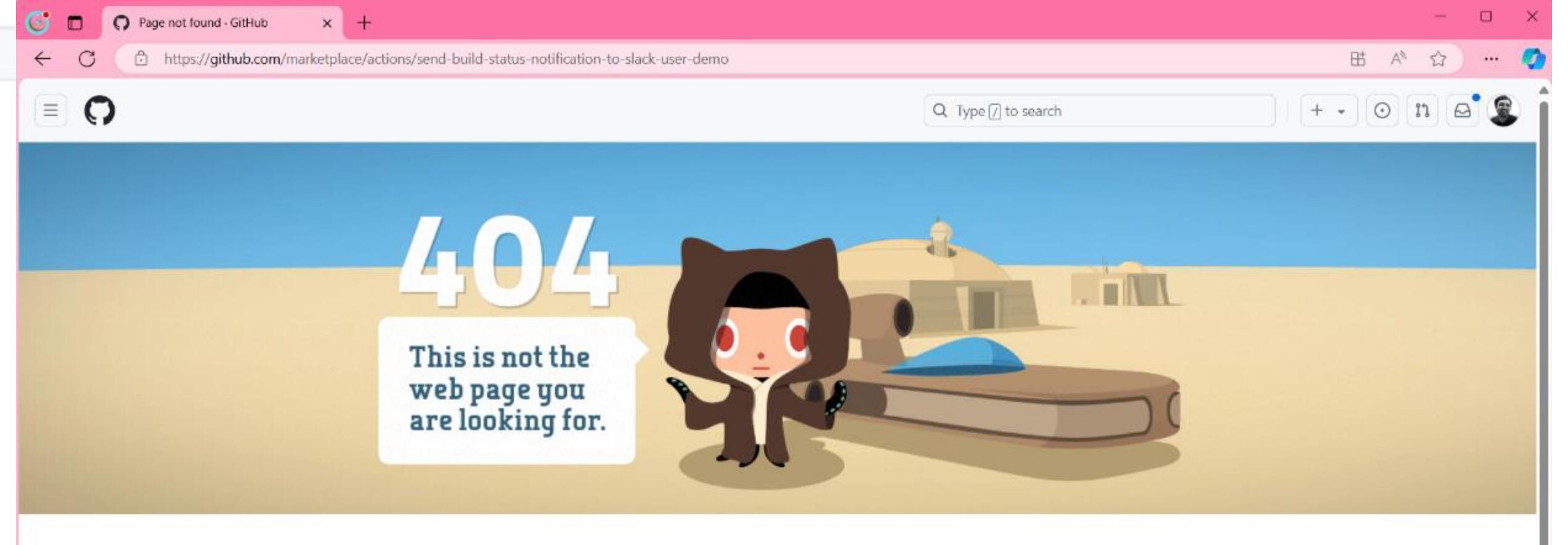
Release Action

Publish this Action to the GitHub Marketplace
Your Action will be discoverable in the Marketplace and available in GitHub search.

v0.4.0 Previous tag: auto Generate release notes

✓ Existing tag

v0.4.0



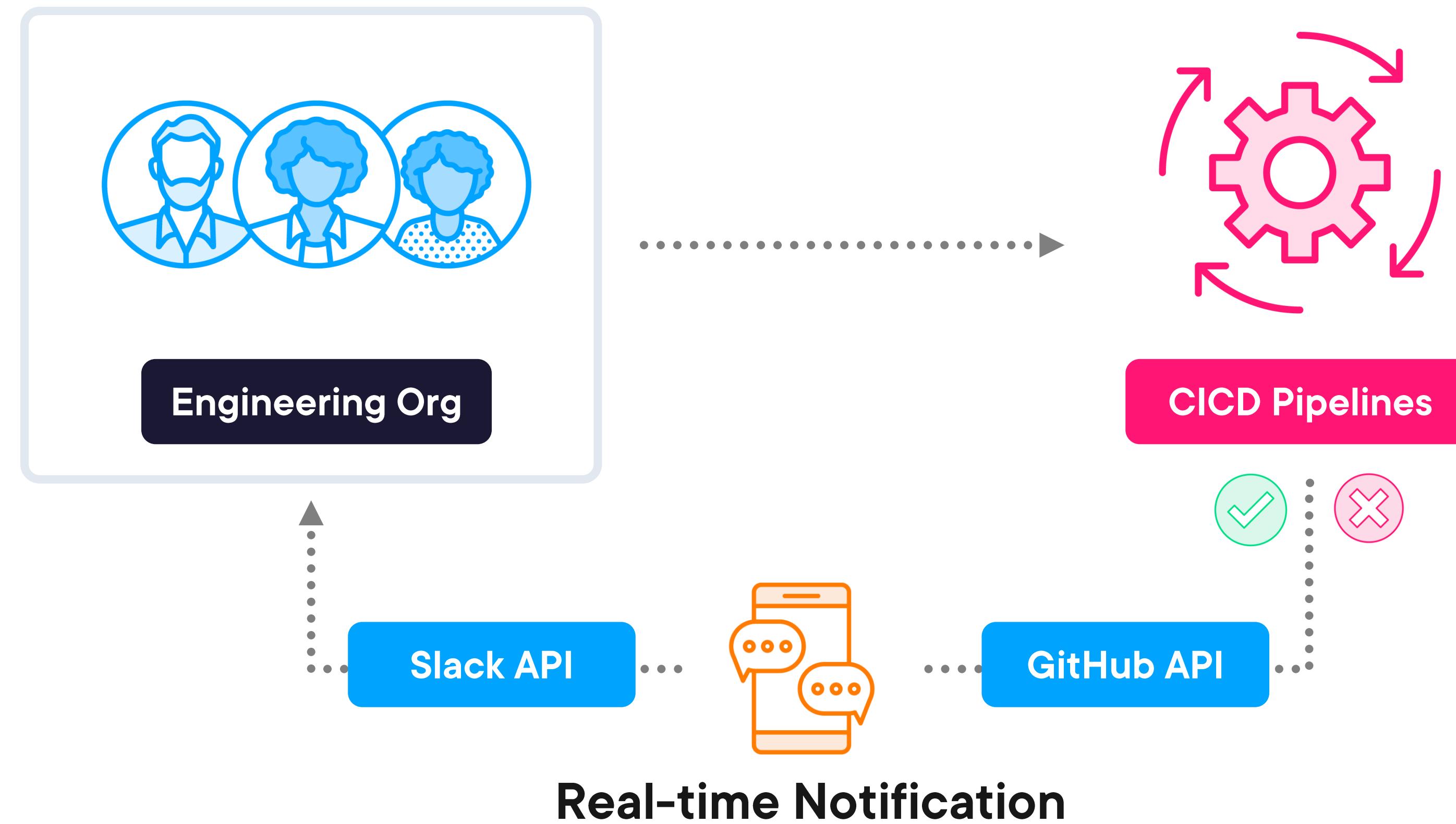
The screenshot shows a GitHub release interface for an action named 'Release Action'. At the top, there are two tabs: 'Releases' (which is selected) and 'Tags'. Below the tabs, there's a section titled 'Release Action' with a checkbox labeled 'Publish this Action to the GitHub Marketplace'. A note below the checkbox states: 'Your Action will be discoverable in the Marketplace and available in GitHub search.' Underneath this are three buttons: a dropdown for the tag version ('v0.4.0'), a button to 'Previous tag: auto', and a 'Generate release notes' button. A checkmark indicates 'Existing tag' and shows 'v0.4.0'. To the right of the release interface is a screenshot of a web browser window. The browser's title bar says 'Page not found - GitHub'. The address bar shows the URL 'https://github.com/marketplace/actions/send-build-status-notification-to-slack-user-demo'. The main content area of the browser displays a 404 error page. The page has a blue header with the number '404' in large white digits. Below it, a speech bubble contains the text 'This is not the web page you are looking for.' There is also a cartoon character of a brown cat-like figure standing in a desert landscape with a small building in the background.



CTO



Applying to CI/CD Pipeline Status Notifier



Summary

Composed multiple action types together using Composite actions

Tested using end-to-end workflows

Added debugging context with workflow commands

Overcame Dockerfile issues and build context passing

Fixed issues passing inputs and outputs between steps

Worked around a real bug in GitHub Actions



**Successfully preparing for a
skill certification means
putting what you learn into
practice.**





Your Homework

Decide on an action idea, design it on paper, and try to publish it to the GitHub Marketplace.



**Good luck and
thanks for
joining me!**

www.linkedin.com/in/kamranayub

