



Github Actions as Publishing Tools

Github actions are a new offering from Github that allows you to automate processes and tooling for Github hosted projects, similar to what you'd get in a CI environment.

The idea is that you create a [YAML](#) file with the list of steps to run and the operating system you want to run them on. It is possible to run an action in multiple operating systems (Windows, Mac and Linux are available).

The following test action checks if the dependencies have been caches and if they are not, it installs the dependencies specified in `package-lock.json` and then runs `gulp`.

```
name: Test
on: [push]

jobs:
  test:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v1
      - uses: actions/cache@v1
        with:
          path: node_modules
          key: ${{ runner.os }}-node-${{ hashFiles('**/package-lock.json') }}
      - name: Install Dependencies
        if: steps.cache.outputs.cache-hit != 'true'
        run: npm install
      - run: gulp
```

If we combine this with [Github pages](#) we now have a full static site publishing solution that will update content everytime we push new content to the repository.

Furthermore we can tie the action to more than one even, for example the

following YAML snippet works when pushing into the master branch and whenever a pull request is closed.

```
name: Test
on:
  push:
    branches:
      - master
  pull_requests:
    types: [closed]
```

Note

I've asked Github support if there is a way to specify merged PRs rather than just closed. I don't know if the closed status implies merged since you can close a PR without merging it.

Research Project: Collaborative Publishing

Now that we have the basics of actions, let's put them together into a working project.

The idea is to take a static site generator, as described in [Static Site Generators: Nunjucks and Gulp](#) with this [code repository](#)

The idea is that only people creating the site can edit the content of the repository but everyone with a Github account can fork the repository and submit content to the repo through [pull requests](#) without having to build the repository every time, unless you want to :-)

Building the workflow

I've chosen to implement the workflow from the repository itself. Create a .github/

```
name: Publish
on:
  push:
    branches:
      - master
  pull_request:
    types: [closed]

jobs:
  publish:
    runs-on: ubuntu-latest

    steps:
      - name: checkout code
        uses: actions/checkout@v1

      - name: Setup Node
        uses: actions/setup-node@v1
        with:
          node-version: '12.x'
          registry-url: 'https://registry.npmjs.org'

      - name: build
        run: |
          npm install
          npm run prepare
          npm run publish

      - name: Deploy
        uses: maxheld83/ghpages@v0.2.1
        with:
          branch: gh-pages
          env.BUILD_DIR: "docs/"
          secrets: "secrets.ACCESS_TOKEN"
```

The workflow uses the following pre-defined actions:

- [checkout](#)
- [setup-node](#)

- [ghpages](#)

There is a large repository of ready-made actions [available on Github](#).

It also leverages the process already in the repository. We run the existing build process from the repository and we don't even need to install the node packages on every developer's machine.

The first version didn't work. I mean, it did work but produced no output so I went back to the drawing board and used an example from [pierresaid](#) at [dev.to](#) as the basis for the example below:

Resources

- [Github Actions](#)
- [Core concepts for GitHub Actions](#)
- [Automating your workflow with GitHub Actions](#)
- [Creating a workflow file](#)
- [Github Actions Node.js starter workflow](#)