Using custom workflows with GitHub Pages

In this article

You can take advantage of using GitHub Actions and GitHub Pages by creating a workflow file or choosing from the predefined workflows.

> GitHub Pages is available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, GitHub Enterprise Cloud, and GitHub Enterprise Server. For more information, see "GitHub's plans."

About custom workflows 🔗

Custom workflows allow GitHub Pages sites to be built via the use of GitHub Actions. You can still select the branch you would like to use via the workflow file, but you are able to do much more with the use of custom workflows. To start using custom workflows you must first enable them for your current repository. For more information, see "Configuring a publishing source for your GitHub Pages site."

Configuring the `configure-pages` action 🔗

GitHub Actions enables the use of GitHub Pages through the `configure-pages` action, which also lets you gather different metadata about a website. For more information, see the `configure-pages` action.

To use the action place this snippet under your `jobs` in the desired workflow.

```
- name: Configure GitHub Pages
  uses: actions/configure-pages@v3
```

This action helps support deployment from any static site generator to GitHub Pages. To make this process less repetitive you can use starter workflows for some of the most widely used static site generators. For more information, see "Using starter workflows."

Configuring the `upload-pages-artifact` action 🔗

The `upload-pages-artifact` actions enables you to package and upload artifacts. The

GitHub Pages artifact should be a compressed `gzip` archive containing a single `tar` file. The `tar` file must be under 10GB in size and should not contain any symbolic or hard links. For more information, see the [upload-pages-artifact](#) action.

To use the action in your current workflow place this snippet under `jobs`.

```
- name: Upload GitHub Pages artifact
  uses: actions/upload-pages-artifact@v1
```

Deploying GitHub Pages artifacts 🔗

The `deploy-pages` action handles the necessary setup for deploying artifacts. To ensure proper functionality, the following requirements should be met:

- The job must have a minimum of `pages: write` and `id-token: write` permissions.
- The `needs` parameter must be set to the `id` of the build step. Not setting this parameter may result in an independent deployment that continuously searches for an artifact that hasn't been created.
- An `environment` must be established to enforce branch/deployment protection rules. The default environment is `github-pages`.
- To specify the URL of the page as an output, utilize the `url:` field.

For more information, see the [deploy-pages](#) action.

```
...

jobs:
  deploy:
    permissions:
      contents: read
      pages: write
      id-token: write
    runs-on: ubuntu-latest
    needs: jekyll-build
    environment:
      name: github-pages
      url: ${{steps.deployment.outputs.page_url}}
    steps:
      - name: Deploy artifact
        id: deployment
        uses: actions/deploy-pages@v1
...
```

Linking separate build and deploy jobs 🔗

You can link your `build` and `deploy` jobs in a single workflow file, eliminating the need to create two separate files to get the same result. To get started on your workflow file, under `jobs` you can define a `build` and `deploy` job to execute your jobs.

```
...

jobs:
  # Build job
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      - name: Setup Pages
        id: pages
```

```yaml
          uses: actions/configure-pages@v3
        - name: Build with Jekyll
          uses: actions/jekyll-build-pages@v1
          with:
            source: ./
            destination: ./_site
        - name: Upload artifact
          uses: actions/upload-pages-artifact@v2

  # Deployment job
  deploy:
    environment:
      name: github-pages
      url: ${{steps.deployment.outputs.page_url}}
    runs-on: ubuntu-latest
    needs: build
    steps:
      - name: Deploy to GitHub Pages
        id: deployment
        uses: actions/deploy-pages@v2
...
```

In certain cases, you might choose to combine everything into a single job, especially if there is no need for a build process. Consequently, you would solely focus on the deployment step.

```yaml
...

jobs:
  # Single deploy job no building
  deploy:
    environment:
      name: github-pages
      url: ${{steps.deployment.outputs.page_url}}
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      - name: Setup Pages
        uses: actions/configure-pages@v3
      - name: Upload Artifact
        uses: actions/upload-pages-artifact@v2
        with:
          # upload entire directory
          path: '.'
      - name: Deploy to GitHub Pages
        id: deployment
        uses: actions/deploy-pages@v2

...
```

You can define your jobs to be run on different runners, sequentially, or in parallel. For more information, see "Using jobs."

Legal