

# Release Workflow

---

The Release Workflow is a GitHub Actions workflow designed to automate the process of creating and uploading releases for a project. This workflow streamlines the release management process by compiling code, packaging artifacts, and creating GitHub releases with associated assets.

- [Features](#)
- [Testing Release](#)

## Features

- **Automated Release Process:** The workflow automates the release process, reducing manual intervention and ensuring consistency in release creation.
- **Versioned Releases:** Each release is associated with a version tag, facilitating version control and tracking of changes over time.
- **Artifact Packaging:** The workflow builds and packages artifacts, such as compiled code or distribution files, to be included in the release.
- **GitHub Release Creation:** The workflow creates GitHub releases, allowing maintainers to document release notes, changelogs, and other relevant information.
- **Release Asset Upload:** Assets, such as compiled binaries or documentation files, are uploaded and associated with the GitHub release, providing users with easy access to release artifacts.

## Testing Release

### 1. Create Workflow File:

- Create a new file named `17-01-Release.yml` in the `.github/workflows` directory of your repository.

```
name: 17-01-Release

on:
  workflow_dispatch

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Setup Java
        uses: actions/setup-java@v4
        with:
          distribution: 'adopt'
          java-version: '11'
          cache: 'maven'

      - name: Build with Maven
```

```

    run: mvn clean install -Dmaven.test.skip=true

  - name: Upload jar to folder
    uses: actions/upload-artifact@v4
    with:
      name: my-artifact
      path: target/

release:
  runs-on: ubuntu-latest
  needs: build

  steps:

    - name: Checkout repository
      uses: actions/checkout@v2

    - name: Download web-app content
      uses: actions/download-artifact@v4
      with:
        name: my-artifact
        path: target/

    - name: View content
      run: ls -R

    - name: Archive site content
      uses: thedoctor0/zip-release@master
      with:
        filename: site.zip

    - name: Create GitHub release
      id: create-new-release
      uses: actions/create-release@v1
      env:
        GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
      with:
        tag_name: ${ github.run_number }
        release_name: Release ${ github.run_number }

    - name: Upload release asset
      uses: actions/upload-release-asset@v1
      env:
        GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
      with:
        upload_url: ${ steps.create-new-release.outputs.upload_url }
        asset_path: ./site.zip
        asset_name: site-v${ github.run_number }.zip
        asset_content_type: application/zip

```

- Copy and paste the provided YAML configuration into this file.

## 2. Understanding the Workflow:

- This workflow is triggered manually via the GitHub UI (`workflow_dispatch`).
- It contains two jobs: `build` and `release`.
- Each job runs on an `ubuntu-latest` runner.
- The `build` job checks out the code, sets up Java, builds the project with Maven, and uploads the resulting artifact to a folder named `target/`.
- The `release` job depends on the `build` job. It checks out the code, downloads the artifact from the `target/` folder, archives the site content into a ZIP file, creates a GitHub release using the `actions/create-release@v1` action, and uploads the ZIP file as a release asset.

### 3. Testing the Workflow:

- Commit and push the workflow file (`17-01-Release.yml`) to your repository.
- Navigate to the "Actions" tab in your GitHub repository.
- Manually trigger the workflow by clicking on the "Run workflow" button for the `17-01-Release` workflow.
- Monitor the workflow run and verify that both jobs (`build` and `release`) complete successfully.
- Check the logs of each job to ensure that the project builds without errors and that the release assets are created and uploaded correctly.
- Check in Release tab in github to verify that release as been added.