
GITHUB ACTIONS DOCUMENTATION

Introduction to GitHub Actions

GitHub Actions is a powerful, flexible, and integrated automation platform provided by GitHub. It allows developers to automate their software development workflows directly within their GitHub repositories. With GitHub Actions, you can build, test, and deploy your code without ever leaving GitHub, streamlining your development process and enabling continuous integration (CI) and continuous delivery (CD) practices.

Key Features

- 1. Workflow Automation:** GitHub Actions enables you to automate any workflow directly within your GitHub repository. Whether it's building, testing, deploying, or any custom process, you can define workflows using YAML syntax, making it easy to understand and maintain.
- 2. Wide Range of Triggers:** Workflows in GitHub Actions can be triggered by various events, such as pushes to a repository, pull requests, issue comments, repository dispatches, scheduled events, and more. This flexibility allows you to customize workflows based on your specific requirements.
- 3. Extensive Ecosystem:** GitHub Actions provides a vast ecosystem of actions, reusable units of code, that you can use in your workflows. These actions cover a wide range of tasks, from common actions like checking out code or running tests to more specialized actions for specific tools and platforms.
- 4. Matrix Builds:** GitHub Actions supports matrix builds, allowing you to run the same workflow across multiple configurations simultaneously. This is particularly useful for testing your code across different operating systems, programming languages, or dependency versions.
- 5. Parallelism and Concurrency:** Workflows in GitHub Actions can run steps in parallel, speeding up the overall execution time. You can also control the concurrency of workflows to ensure that they don't overload your resources or external services.

6. Secrets Management: GitHub Actions provides a secure way to store and use sensitive information, such as API tokens, SSH keys, or environment variables, using GitHub Secrets. These secrets can be securely accessed by your workflows without exposing them in your repository.

7. Integration with GitHub: Since GitHub Actions is tightly integrated with GitHub, you can easily view and manage your workflows directly within the GitHub UI. This includes monitoring workflow runs, viewing logs, and configuring workflow triggers and settings.

Using GitHub Actions for Continuous Integration (CI)

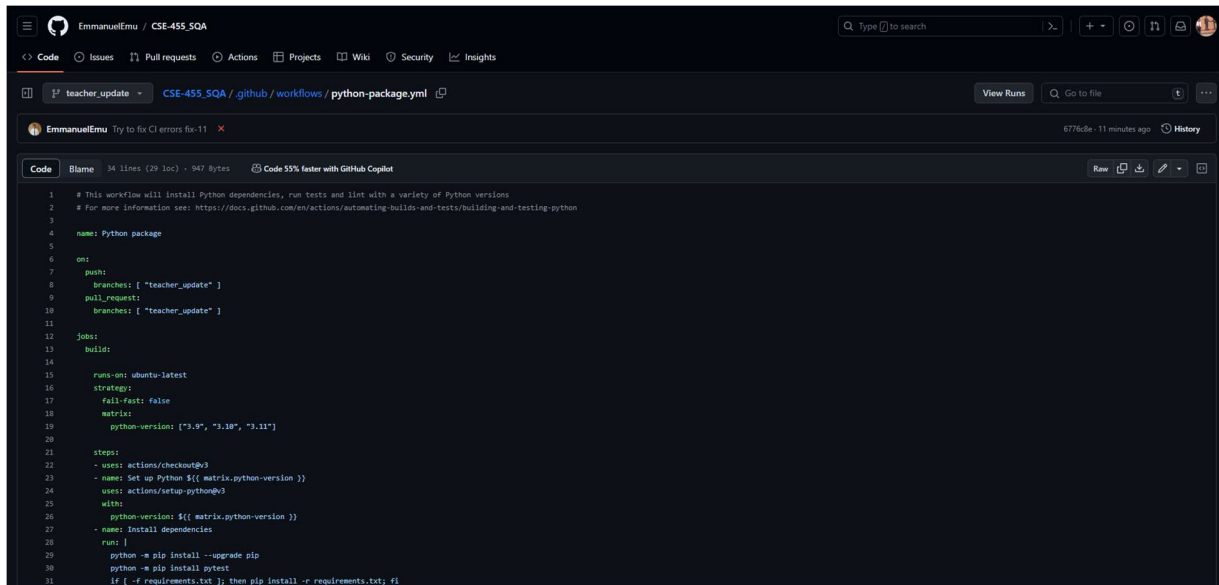
Continuous Integration (CI) is a software development practice where code changes are automatically built, tested, and verified whenever they are integrated into a shared repository. GitHub Actions provides a seamless way to implement CI workflows for your projects. Here's how you can use GitHub Actions for CI:

1. Defining CI Workflows: Create a `.github/workflows` directory in your repository and define YAML files to define your CI workflows. These files typically specify the events that trigger the workflow (e.g., pushes, pull requests) and the sequence of steps to be executed.



2. Setting Up Jobs and Steps: Within your workflow file, define one or more jobs, each representing a set of related tasks to be performed. Each job consists of a series of steps, which are individual units of work, such

as checking out code, installing dependencies, running tests, and deploying artifacts.



```
1 # This workflow will install Python dependencies, run tests and lint with a variety of Python versions
2 # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-python
3
4 name: Python package
5
6 on:
7   push:
8     branches: [ "teacher_update" ]
9   pull_request:
10    branches: [ "teacher_update" ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16     strategy:
17       fail-fast: false
18     matrix:
19       python-version: ["3.9", "3.10", "3.11"]
20
21     steps:
22     - uses: actions/checkout@v3
23     - name: Set up Python ${{ matrix.python-version }}
24       uses: actions/setup-python@v3
25     - with:
26       python-version: ${{ matrix.python-version }}
27     - name: Install dependencies
28       run: |
29         python -m pip install --upgrade pip
30         python -m pip install pytest
31         if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
```

3. Configuring Triggers: Configure triggers for your workflows to specify when they should be executed. This could include triggers based on specific branches, tags, or types of events (e.g., pull_request, push).

4. Using Actions and Services: Leverage the vast ecosystem of GitHub Actions and third-party actions to perform common tasks in your workflows. These actions encapsulate reusable logic and can be easily integrated into your workflows.

5. Running Tests: Use your CI workflows to run automated tests on your codebase whenever changes are made. This helps catch bugs early and ensures that your codebase remains in a functional state.

Actions

New workflow

All workflows

Workflows

Django CI

Python package

Management

Caches

Runners

All workflows

Showing runs from all workflows

Filter workflow runs

65 workflow runs

Event

Status

Branch

Actor

Adding CI testing report to Resources folder

Django CI #41: Commit 2b4492f pushed by EmmanuelEmu

main

now

in progress

...

Adding CI testing report to Resources folder

Python package #26: Commit 2b4492f pushed by EmmanuelEmu

main

now

in progress

...

Try to fix CI errors fix-11

Python package #25: Commit 6776c8e pushed by EmmanuelEmu

teacher_update

5 minutes ago

1m 10s

...

Try to fix CI errors fix-11

Django CI #40: Commit 6776c8e pushed by EmmanuelEmu

teacher_update

5 minutes ago

1m 1s

...

Try to fix CI errors fix-10

Django CI #39: Commit 0cd99d pushed by EmmanuelEmu

teacher_update

7 minutes ago

55s

...

Try to fix CI errors fix-10

Python package #24: Commit 0cd99d pushed by EmmanuelEmu

teacher_update

7 minutes ago

1m 7s

...

fixing cicd error again

Django CI #38: Commit 7177166 pushed by kibria282096

update_student

11 minutes ago

1m 6s

...

fixing cicd error again

Python package #23: Commit 7177166 pushed by kibria282096

update_student

11 minutes ago

1m 6s

...

fixing cicd workflows error

Django CI #37: Commit 76d7abd pushed by kibria282096

update_student

16 minutes ago

1m 4s

...

fixing cicd workflows error

Python package #22: Commit 76d7abd pushed by kibria282096

update_student

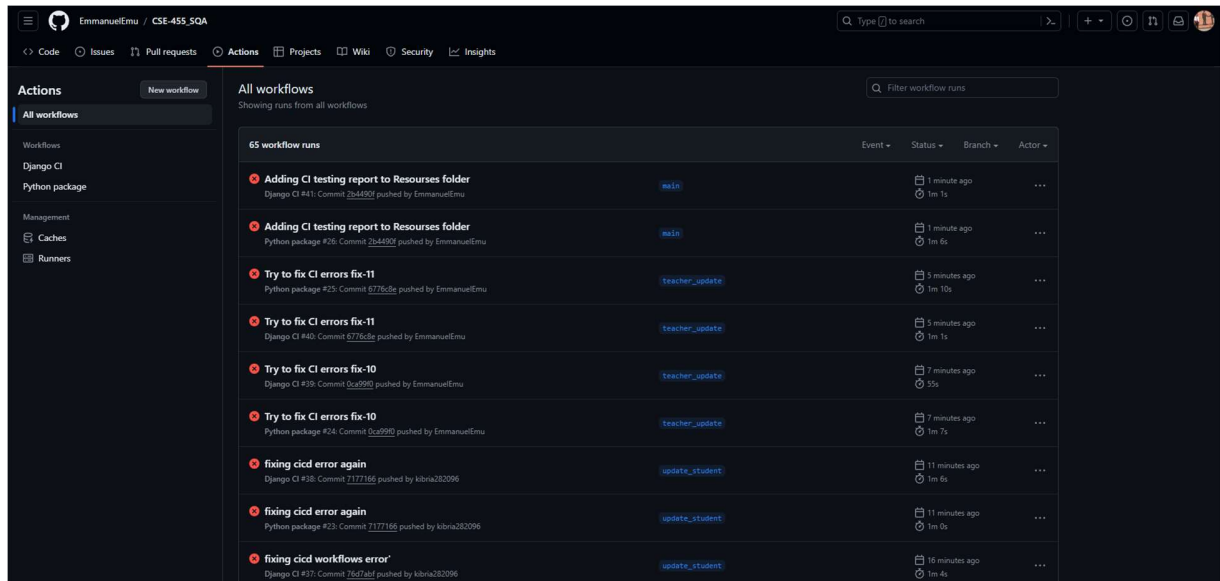
16 minutes ago

1m 4s

...

6. Monitoring and Notifications: Monitor the status of your CI workflows directly within GitHub. GitHub provides detailed logs and status indicators for each workflow run, allowing you to quickly identify any issues.

You can also configure notifications to alert you of workflow failures or other important events.



The screenshot shows the GitHub Actions interface for the repository 'EmmanuelEmu / CSE-455_SQA'. The left sidebar contains navigation links: Code, Issues, Pull requests, Actions (selected), Projects, Wiki, Security, and Insights. Under the 'Actions' tab, there are links for 'All workflows', 'Workflows', 'Django CI', 'Python package', 'Management', 'Caches', and 'Runners'. The main content area is titled 'All workflows' and shows a list of workflow runs. The list has columns for 'Event', 'Status', 'Branch', and 'Actor'. The runs are listed in descending order of time, with the most recent run at the top. Each run entry includes a red circle icon with a white 'x' indicating a failed status, the workflow name, the commit hash, the branch, the status, the actor, and the time ago.

Event	Status	Branch	Actor
Adding CI testing report to Ressources folder	main	1 minute ago	...
Adding CI testing report to Ressources folder	main	1 minute ago	...
Try to fix CI errors fix-11	teacher_update	3 minutes ago	...
Try to fix CI errors fix-11	teacher_update	3 minutes ago	...
Try to fix CI errors fix-10	teacher_update	7 minutes ago	...
Try to fix CI errors fix-10	teacher_update	7 minutes ago	...
fixing cicd error again	update_student	11 minutes ago	...
fixing cicd error again	update_student	11 minutes ago	...
fixing cicd workflows error	update_student	16 minutes ago	...

7. Integrating with External Services: GitHub Actions can integrate with external services, such as cloud providers, code quality tools, or deployment platforms, to automate additional tasks in your CI process. This enables you to create end-to-end CI/CD pipelines tailored to your specific requirements.