

Add Continuous Integration with Linting



Estimated time needed: 60 minutes

Your team is growing! Management has decided to hire front end and backend engineers to ensure features on the roadmap are developed in time for future releases. However, this means that multiple engineers will need to work in parallel on the repository. You are tasked to ensure the code being pushed to the main branch meets the team coding style and is free of syntax errors. In this lab, you will add linting to your repository that automatically checks for such errors whenever a developer creates a pull request or whenever a branch is merged into the default main branch. Before we dive into the lab, here is a primer on GitHub Actions.

GitHub Actions

GitHub actions provide an event-driven way to automate tasks in your project. There are several kinds of events you can listen to. Here are a few examples:

- **push:** Runs tasks when someone pushes to a repository branch.
- **pull_request:** Runs tasks when someone creates a pull request. You can also start tasks when certain activities happen, such as:
 - PR opened
 - PR closed
 - PR reopened
- **create:** Run tasks when someone creates a branch or a tag.
- **delete:** Run tasks when someone deletes a branch or a tag.
- **manually:** Jobs are kicked off manually.

GitHub Action Components

You will use one or more of the following components in this lab:

- **Workflows:** Collection of jobs you can add to your repository.
- **Events:** Activity that launches a workflow.
- **Jobs:** Sequence of one or more steps. Jobs are run in parallel by default.
- **Steps:** Individual tasks that can run in a job. A step may be an action or a command.
- **Actions:** Smallest block of a workflow.

GitHub Workflow

We have provided a workflow template in this file: `.github/workflows/linter.yml`. Let's examine this file.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
16. 16
17. 17
18. 18
19. 19
20. 20
21. 21
22. 22
23. 23
24. 24
25. 25
26. 26
27. 27
28. 28
29. 29
30. 30
31. 31
32. 32
33. 33
34. 34
35. 35
36. 36
37. 37
38. 38
39. 39
40. 40
41. 41
42. 42
43. 43
```

44. 44
45. 45
46. 46
47. 47
48. 48
49. 49
50. 50
51. 51
52. 52
53. 53
54. 54
55. 55
56. 56
57. 57
58. 58
59. 59
60. 60
61. 61
62. 62

```
1. name: 'Lint Code'
2.
3. on:
4.   push:
5.     branches: [master, main]
6.   pull_request:
7.     branches: [master, main]
8.
9. jobs: # list of things to do
10.  lint_function_js:
11.    name: Linting JavaScript Function
12.    runs-on: ubuntu-latest
13.    steps: # list of steps
14.      - name: Install NodeJS
15.        uses: actions/setup-node@v3
16.        with:
17.          node-version: 16
18.      - name: Code Checkout
19.        uses: actions/checkout@v3
20.      - name: Install Dependencies
21.        run: npm install
22.        working-directory: functions/sample/nodejs
23.      - name: Code Linting
24.        run: npm run lint
25.        working-directory: functions/sample/nodejs
26.
27.
28.  lint_function_python:
29.    name: Linting Python Function
30.    runs-on: ubuntu-latest
31.    steps:
32.      - uses: actions/checkout@v3
33.      - name: Set up Python 3.7
34.        uses: actions/setup-python@v4
35.        with:
36.          python-version: 3.7
37.      - name: Install dependencies
38.        run: |
39.          python -m pip install --upgrade pip
40.          pip install pylint
41.          if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
42.      - name: Lint with Pylint
43.        run: pylint *.py
44.        working-directory: functions/sample/python
45.
46.  lint_django_server:
47.    name: Linting Django Server
48.    runs-on: ubuntu-latest
49.    steps:
50.      - uses: actions/checkout@v3
51.      - name: Set up Python 3.7
52.        uses: actions/setup-python@v4
53.        with:
54.          python-version: 3.7
55.      - name: Install dependencies
56.        run: |
57.          python -m pip install --upgrade pip
58.          pip install pylint
59.          if [ -f requirements.txt ]; then pip install -r requirements.txt; fi
60.      - name: Lint with Pylint
61.        run: pylint *.py
62.        working-directory: server
```

Copied!

1. The first line names the workflow.
2. The next line defines when this workflow will run. We want the workflow to run when developers push a change to the main branch or create a PR. We also want them to be able to run the workflow manually. These three ways are captured as follows:
 - o run manually: `workflow_dispatch`:
 - o run on push to the main branch (main or master):

```
1. 1
2. 2
1. push:
2.   branches: [master, main]
```

Copied!

- run when PR is created on main branches (main or master):

```
1. 1
2. 2
1. pull_request:
2. branches: [master, main]
```

Copied!

3. We then define all the jobs. We have three jobs in this workflow:
 - `lint_function_js`: Linting JavaScript Function
 - `lint_function_python`: Linting Python Function
 - `lint_django_server`: Linting Django Server

GitHub Jobs

Let's look at each of these jobs:

1. `lint_function_js`
 1. Set up Node.js runtime for the action to run using the `actions/setup-node@v3` action
 2. Check out the code using the `actions/checkout@v3` action.
 3. Install all dependencies using `npm npm install`
 4. Run the linting command defined in `package.json` as `npm run lint` in the `functions/sample/nodejs` directory.
2. `lint_function_python`
 1. Set up Python runtime for the action to run using the `actions/setup-python@v4` action
 2. Check out the code using the `actions/checkout@v3` action.
 3. Install all dependencies using `pip install`
 4. Run the linting command `pylint *.py` in the `functions/sample/python` directory.
3. `lint_django_server`
 1. Set up Python runtime for the action to run using the `actions/setup-python@v4` action
 2. Check out the code using the `actions/checkout@v3` action.
 3. Install all dependencies using `pip install`
 4. Run the linting command `pylint *.py` in the `server` directory.

Enable GitHub Actions

1. To enable GitHub action, log into GitHub and open your forked repo. Next, go to the Actions tab and click on I understand my workflows, go ahead and enable them.

upkarliddar /

agfzb-CloudAppDevelopment_Capstone

forked from [ibm-developer-skills-network/agfzb-CloudAppDevelopment_Capstone](#)

<> Code

🔗 Pull requests

▶ Actions

📁 Projects

📖 Wiki



Workflows aren't being run on this fork

Because this repository contained workflow files when it was forked, we have disabled workflows on this fork. Make sure you understand the configured workflows and their expected behavior before running them on a forked repository.

I understand my workflows, go ahead and run them

[View the workflows directory](#)

{ width=800}

2. You should see the `Lint Code` workflow on the left hand panel. Notice that no workflows have been run yet, as indicated in the center of the screen.

Actions

New workflow

All workflows

Lint Code

Management

 Caches

All workflows

Showing runs from all workflows

0 workflow runs


{ width=800}


Run the workflow


Let's run the workflow now.


1. Open the Actions tab again and click on the Lint Code action. Then, click on Run workflow dropdown and the Run workflow button.

<> Code

 Pull requests

 **Actions**

 Projects

 Wiki


Actions

New workflow

All workflows

Lint Code

Management

 Caches

Lint Code

linter.yml

0 workflow runs

This workflow has a wor

{ width=800}


2. You should see a new workflow run in the same window. If you don't see anything, click on Actions tab again to refresh the page.

Workflow run was successfully requested.

 **captainfedoraskillup / agfzb-CloudAppDevelopment_**

forked from [ibm-developer-skills-network/agfzb-CloudAppDevelopment_Capstone](#)

 Code

 Pull requests

 **Actions**

 Projects

 Wiki

Actions

New workflow

All workflows

 **Lint Code**

Management

 Caches

Lint Code

[linter.yml](#)

0 workflow runs

This workflow has a wor

 **Lint Code**

Lint Code #18: Manual

{ width=800}

3. You can click on the workflow run to see the individual jobs and the logs for each job.

<> Code

🔗 Pull requests

🎮 Actions

📁 Projects

📖 Wiki

← Lint Code

✔ Lint Code #18

🏠 Summary

Jobs

- ✔ Linting JavaScript Function
- ✔ Linting Python Function
- ✔ Linting Django Server

Run details

🕒 Usage

📄 Workflow file

Linting JavaScript Function

succeeded 2 minutes ago in

- > ✔ Set up job
- > ✔ Install NodeJS
- > ✔ Code Checkout
- > ✔ Install Dependencies
- > ✔ Code Linting
- > ✔ Post Code Check
- > ✔ Post Install Node
- > ✔ Complete job

{ width=800}

Submission

Copy the URL of your forked repo to submit for peer review.

Summary

In this lab, you added a linting service to your application. As a result, all new code will automatically get checked for syntax errors and this will ensure all developers are following the team coding guidelines.

Author(s)

Upkar Lidder

Other Contributor(s)

Yan Luo

Priya

Changelog

Date	Version	Changed by	Change Description
2021-01-28	1.0	Upkar Lidder	Created new instructions for Capstone project
2022-09-13	1.1	K Sundararajan	Updated instructions for clarity
2022-11-18	1.1	CF	removed cd and added linting

© IBM Corporation 2021. All rights reserved.