11/27/23, 10:56 PM about:blank

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 ropened
- 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 once or more steps. Jobs are run in parallel by default.
 Steps: Individual tasks that can run in a job. A step my 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. name: 'Lint Code' 2.
3. on:
4. push:
5. branches: [master, main]
6. pull_request:
7. branches: [master, main] jobs: # list of things to do
lint_function_js:
name: Linting lawsecript Function
runs-on: ubuntu-latest
steps: # list of steps
name: install NodeJS
ust historion/setup-node@v3
with: close/setup-node@v3 node-version: 16 - name: Code Checkout uses: actions/checkout@v3

```
    name: Install Dependencies
run: npm install
working-directory: functions/sample/nodejs
name: Code Linting
run: npm run lint

                          working-directory: functions/sample/nodejs
              lint_function_python:
name: Linting Python Function
runs-on: ubuntu-latest
steps:
                    - uses: actions/checkout@v3
- name: Set up Python 3.7
uses: actions/setup-python@v4
                   with:
python-version: 3.7
name: Install dependencies
run:
python -m plp install --upgrade pip
pip install pylint
if [-f requirements.txt]; then pip install -r requirements.txt; fi
name: Lint with Pylint
run: pylint *.py
writing-directory; functions/sample/python
            lint_django_server
name: Linting Django Server
runs-on: dubtut-latest
see uses: actions/checkoutgw3
- name: Set up Python 3.7
uses: actions/setup-python@w4
kiththon-version: 3.7
- name: Install dependencies
run!
   56.
57.
58.
59.
60.
61.
                     - name: install deprementary
run: |
python -m pip install --upgrade pip
pip install pylint
if {-f requirements.txt}; then pip install -r requirements.txt; fi
-name: Lint with Pylint
                          run: pylint *.py
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:

    run manually: workflow_dispatch:
    run on push to the main branch (main or master):

                    1. 1
2. 2
1. push:
2. branches: [master, main]
                o run when PR is created on main branches (main or master):
                     2. 2
1. pull_request:
2. branches: [master, main]
Copied!

    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

    Set up Node.js runtime for the action to run using the actions/setup-node@v3 action
    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. Int function python

1. Set up Python muttime 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.

    lint_django_server
    Set up Python runtime for the action to run using the actions/setup-python@v4 action

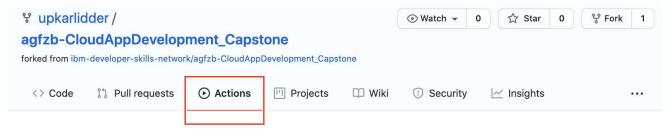
2. Check out the code using the actions/checkout@v3 action.

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.





Workflows aren't being run on this forked repository

Because this repository contained workflow files when it was forked, we have disabled them from running on this fork.

Make sure you understand the configured workflows and their expected usage before enabling Actions on this repository.

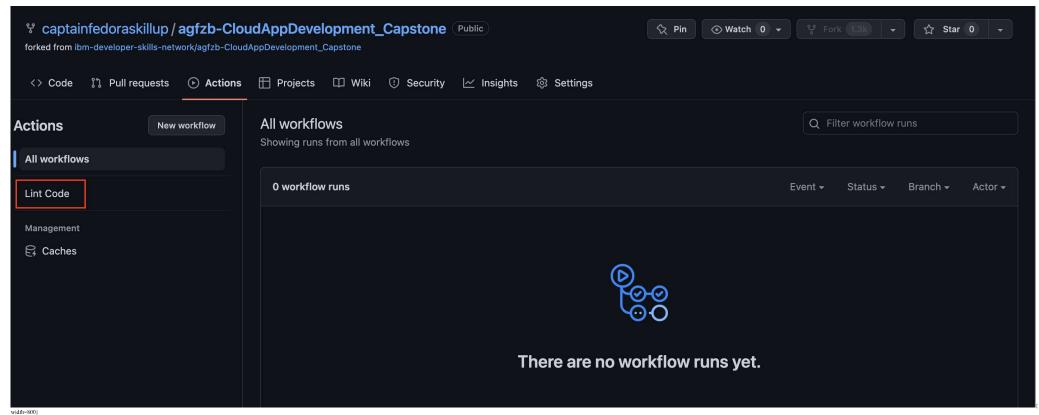
I understand my workflows, go ahead and enable them

View the workflows directory

{ width=800}

about:blank 3/8

^{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.

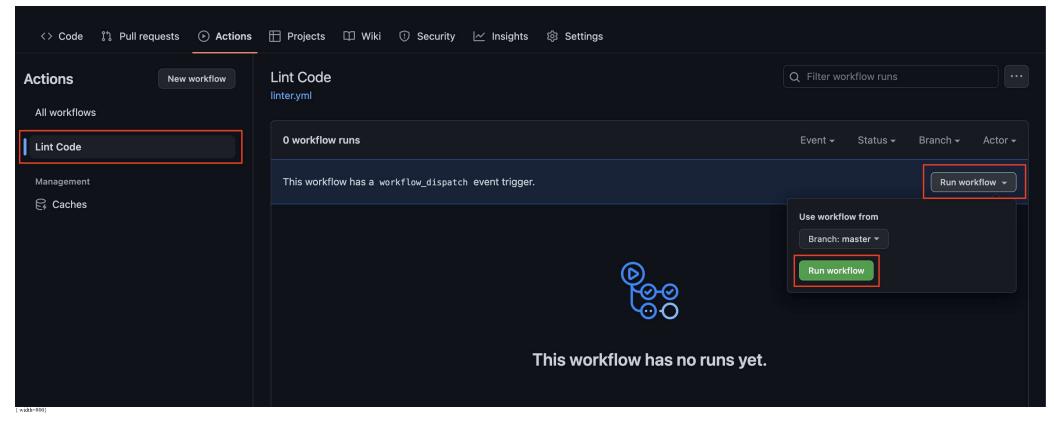


Run the workflow

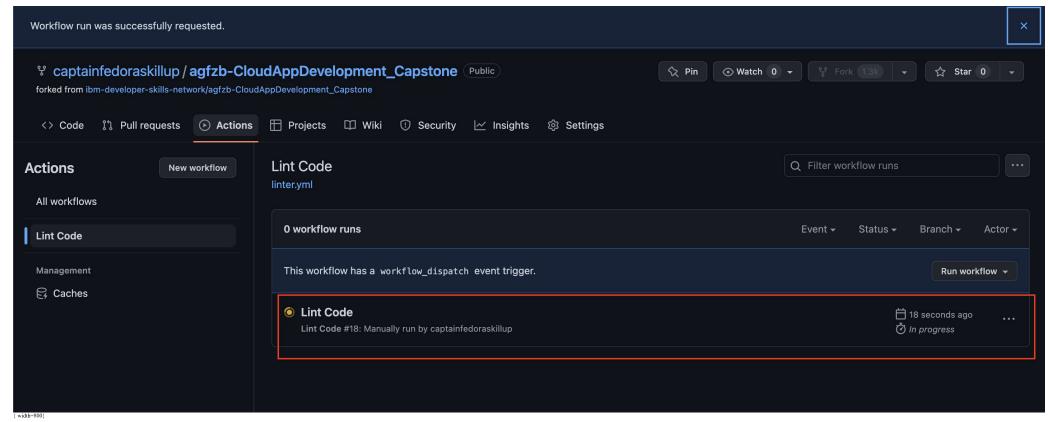
Let's run the workflow now.

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

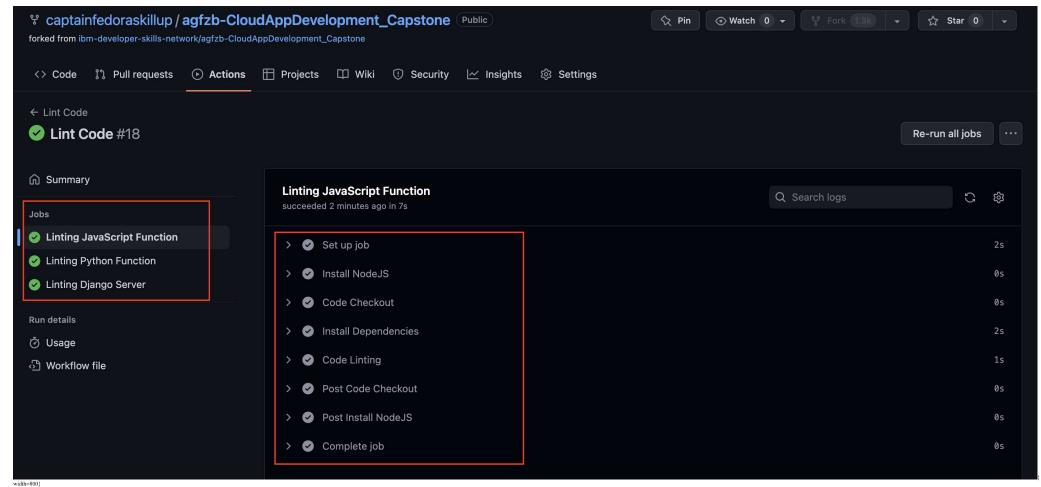
11/27/23, 10:56 PM about:blank



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.



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



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 ed and added linting

© IBM Corporation 2021. All rights reserved.