

Johnson Nguyen

CS 472

Dr. Businge

<https://github.com/nguyenj21/cs472-group4>

Workflow Lab Report

Task 1 & 2 Executive Summary:

- Created a GitHub Actions workflow by adding a .github/workflows directory.
- Inside that directory, created a YAML file named main.yml.
- The workflow is named CI workflow.
- Triggered by pushes to the main branch and pull requests targeting the main branch.
- The workflow runs on an Ubuntu environment.
- Uses the Python 3.9-slim container image.
- Used nosetests to report on coverage
- The workflow consists of the following steps:
 - Checkout: Fetches the code from the repository.
 - Install dependencies: Upgrades pip and installs dependencies from requirements.txt.
 - Lint with flake8: Runs flake8 with specific options for code linting.
 - Run unit tests with nose: Executes unit tests using the nose testing framework.

RED PHASE:

```
Terminal Local x + v
nguyen@DESKTOP-KEQQTRO MINGW64 ~/Desktop/CS472/tddnew/tdd (main)
$ nosetests

Counter tests
- It should create a counter
- Delete a counter (FAILED)
- It should return an error for duplicates
- Read a counter
- Update a counter

=====
FAIL: Delete a counter
-----
Traceback (most recent call last):
  File "C:\Users\Nguyen\Desktop\CS472\tddnew\tdd\tests\test_counter.py", line 84, in test_delete_counter
    self.assertEqual(result.status_code, status.HTTP_204_NO_CONTENT)
AssertionError: 405 != 204
-----
>> begin captured logging << -----
src.counter: INFO: Request to create counter: delete
-----
>> end captured logging << -----

Name          Stats    Miss  Cover   Missing
-----
src\counter.py    25      0  100%
src\status.py      6      0  100%
-----
TOTAL              31      0  100%
-----
Ran 5 tests in 0.220s

FAILED (failures=1)
```

```
1
def test_delete_counter(self):
    """Delete a counter"""
    # creating a counter
    result = self.client.post('/counters/delete')
    # ensure returned successful return code
    self.assertEqual(result.status_code, status.HTTP_201_CREATED)

    # read counter
    result = self.client.delete('/counters/delete')
    self.assertEqual(result.status_code, status.HTTP_204_NO_CONTENT)
    # counterValue = result.get_json()['read']
    # self.assertEqual(baselineValue + 1, updateValue)
    result = self.client.delete('/counters/empty')
    self.assertEqual(result.status_code, status.HTTP_404_NOT_FOUND)
```

Test Case Name: test_delete_counter

Description: This test case verifies the functionality of deleting a counter using the DELETE request.

- **Steps:**

- **Step 1:** Create a counter by sending a POST request to /counters/delete.

- Expected outcome: The response status code should be 201 CREATED.

- **Step 2:** Delete the counter by sending a DELETE request to /counters/delete.

- Expected outcome: The response status code should be 204 NO CONTENT.

- **Step 3:** Attempt to delete a non-existent counter by sending a DELETE request to /counters/empty.

- Expected outcome: The response status code should be 404 NOT FOUND.

GREEN PHASE:

```
@app.route('/counters/<name>', methods=['DELETE'])
def delete_counter(name):
    """Delete a counter"""
    app.logger.info(f"Request to delete counter: {name}")
    global COUNTERS
    if name not in COUNTERS:
        return {"Message": f"Counter {name} not found"}, status.HTTP_404_NOT_FOUND

    COUNTERS[name] += 1
    return {name: COUNTERS[name]}, status.HTTP_204_NO_CONTENT
```

```
Nguyen@DESKTOP-KEQI90 MINGW64 ~/Desktop/CS472/tddnew/tdd (main)
$ nosetests

Counter tests
- It should create a counter
- Delete a counter
- It should return an error for duplicates
- Read a counter
- Update a counter

Name           Stmts  Miss  Cover   Missing
-----
src\counter.py   32      0  100%
src\status.py    6      0  100%
-----
TOTAL            38      0  100%
-----
Ran 5 tests in 0.215s

OK
```

- **Route:** /counters/<name>
- **HTTP Method:** DELETE
- **Functionality:** Deletes a counter with the specified name.
- **Behavior:**
 - If the counter exists (i.e., name is in COUNTERS), it increments the counter value by 1.
 - Returns a response with the updated counter value and a status code of 204 NO CONTENT.
 - If the counter does not exist, returns a message indicating that the counter was not found with a status code of 404 NOT FOUND.

REFACTOR PHASE:

- Coverage is already 100%

```
Name           Stmts  Miss  Cover   Missing
-----
src\counter.py   32      0  100%
src\status.py    6      0  100%
-----
TOTAL            38      0  100%
-----
Ran 5 tests in 0.215s

OK
```