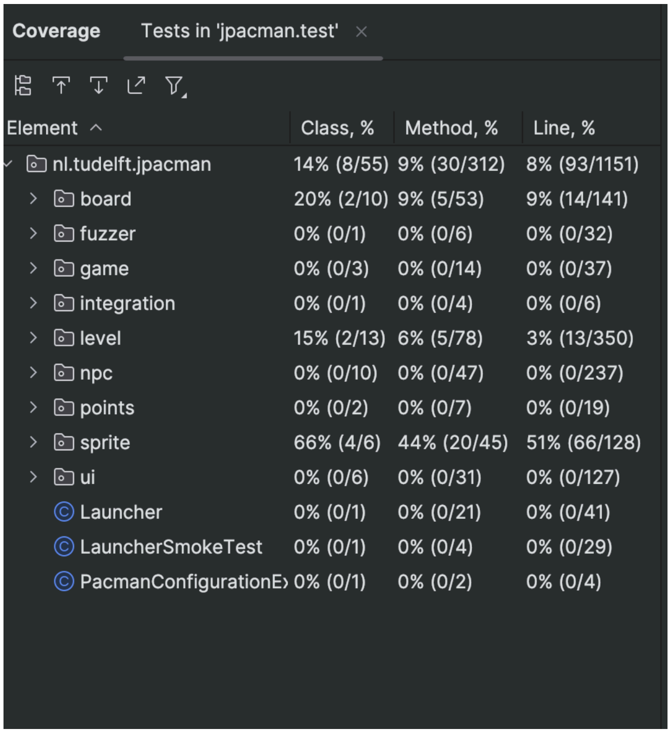
Link to repository:

https://github.com/FPSwitch/samerunittesting

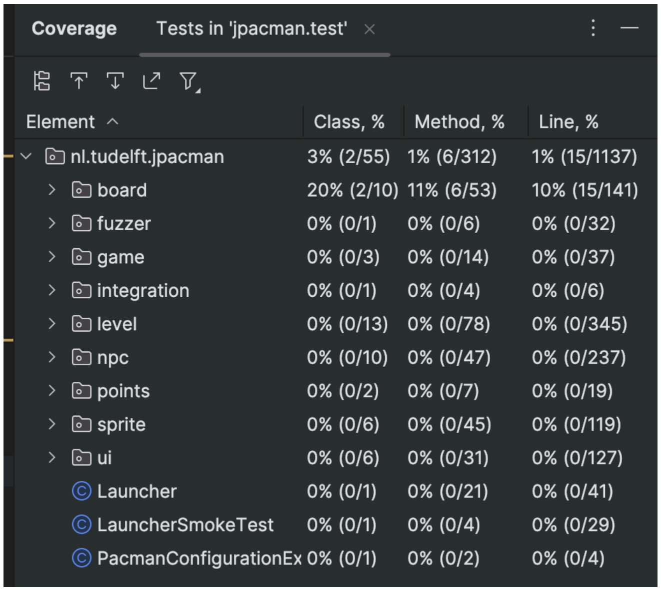
Task 2:

Before Test Coverage:



Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	14% (8/55)	9% (30/312)	8% (93/1151)
board	20% (2/10)	9% (5/53)	9% (14/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2/13)	6% (5/78)	3% (13/350)
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4/6)	44% (20/45)	51% (66/128)
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationE›	0% (0/1)	0% (0/2)	0% (0/4)

After Test Coverage:



Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	3% (2/55)	1% (6/312)	1% (15/1137)
board	20% (2/10)	11% (6/53)	10% (15/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	0% (0/13)	0% (0/78)	0% (0/345)
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	0% (0/6)	0% (0/45)	0% (0/119)
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationEx	0% (0/1)	0% (0/2)	0% (0/4)

Unit Tests:

PlayerTestMoveDown.java:

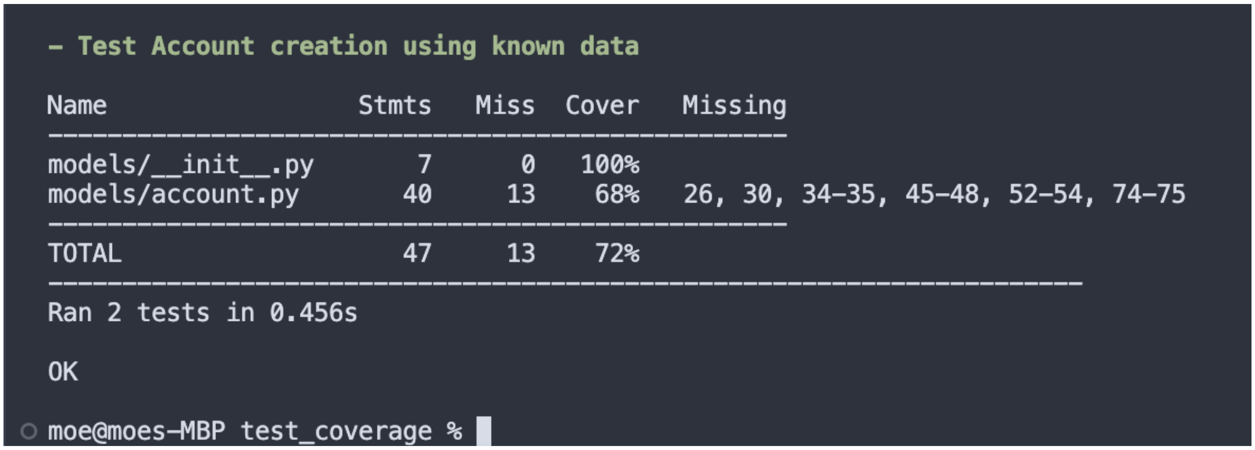
PlayerTestMoveLeft.java:

PlayerTestMoveRight:

Task 3:

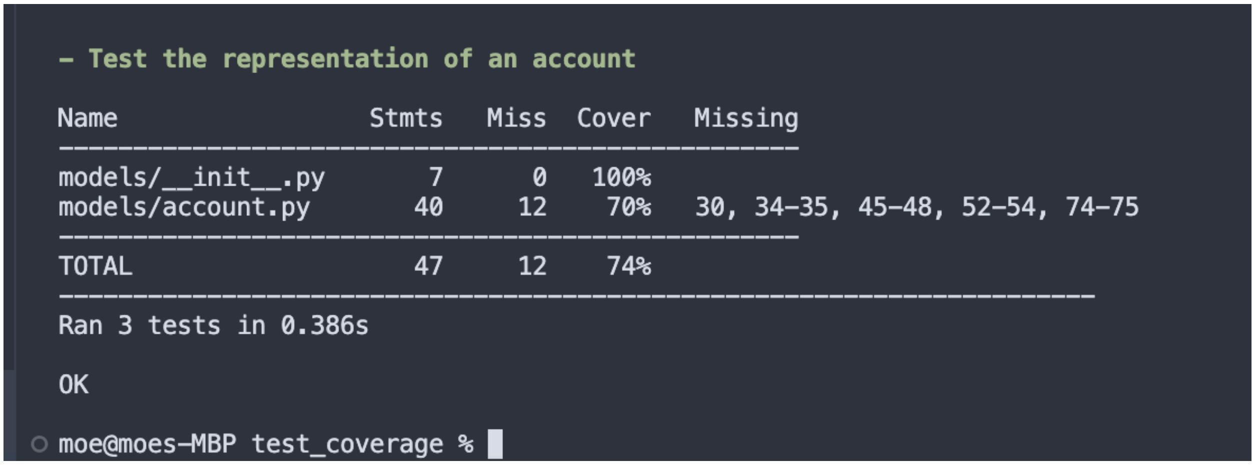
- Are the coverage results from JaCoCo similar to the ones you got from IntelliJ in the last task? Why so or why not?
    -
- Did you find helpful the source code visualization from JaCoCo on uncovered branches?
    -
- Which visualization did you prefer and why? IntelliJ’s coverage window or JaCoCo’s report?
    -

Task 4:

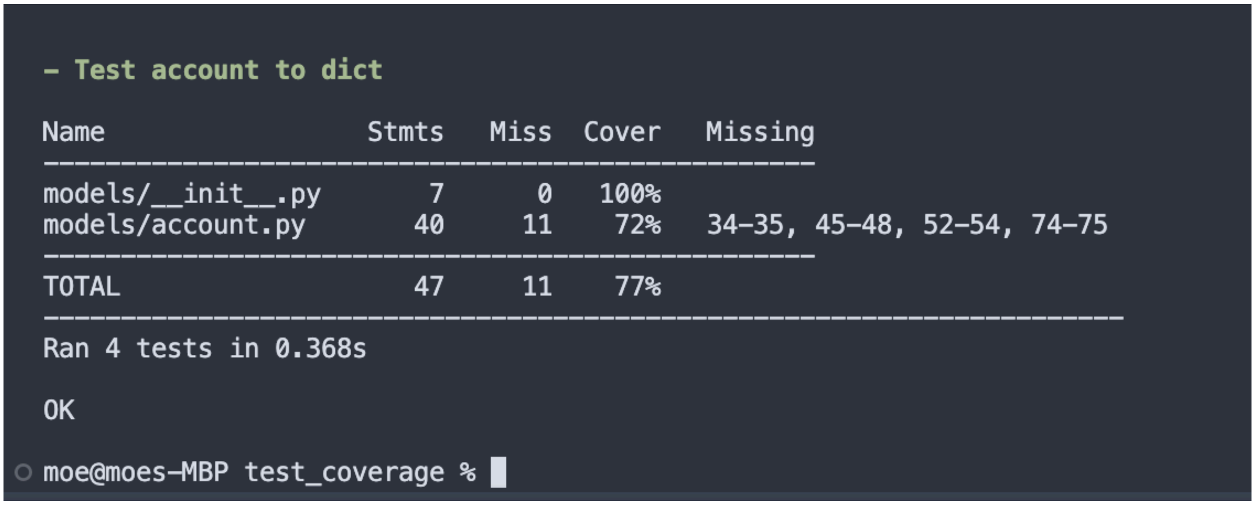


```
- Test Account creation using known data

Name                    Stmts   Miss  Cover   Missing
-----------------------------------------------------
models/__init__.py          7      0   100%
models/account.py          40     13    68%   26, 30, 34-35, 45-48, 52-54, 74-75
-----------------------------------------------------
TOTAL                      47     13    72%
----------------------------------------------------------------------
Ran 2 tests in 0.456s

OK

○ moe@moes-MBP test_coverage %
```

From the start, we had a total coverage of 72%. From here we added a test case for *repr(self)* and *to_dict(self)* into the test_account.py file. The purpose of this was to utilize and call the functions that were defined in the account.py file as this resulted in an increase in coverage by 2% and then an additional 3% going up to **77%.**



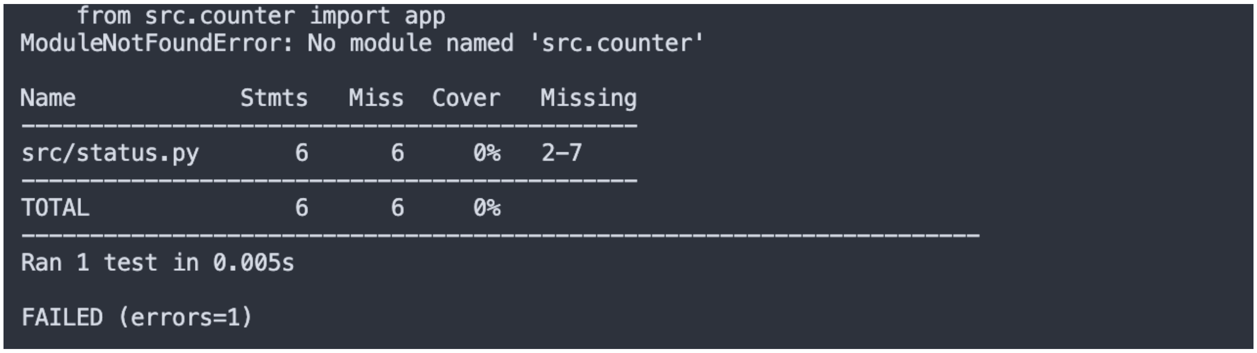
```
- Test the representation of an account

Name                    Stmts   Miss  Cover   Missing
-----------------------------------------------------
models/__init__.py          7      0   100%
models/account.py          40     12    70%   30, 34-35, 45-48, 52-54, 74-75
-----------------------------------------------------
TOTAL                      47     12    74%
----------------------------------------------------------------------
Ran 3 tests in 0.386s

OK

○ moe@moes-MBP test_coverage %
```



```
- Test account to dict

Name                    Stmts   Miss  Cover   Missing
-----------------------------------------------------
models/__init__.py          7      0   100%
models/account.py          40     11    72%   34-35, 45-48, 52-54, 74-75
-----------------------------------------------------
TOTAL                      47     11    77%
----------------------------------------------------------------------
Ran 4 tests in 0.368s

OK

○ moe@moes-MBP test_coverage %
```

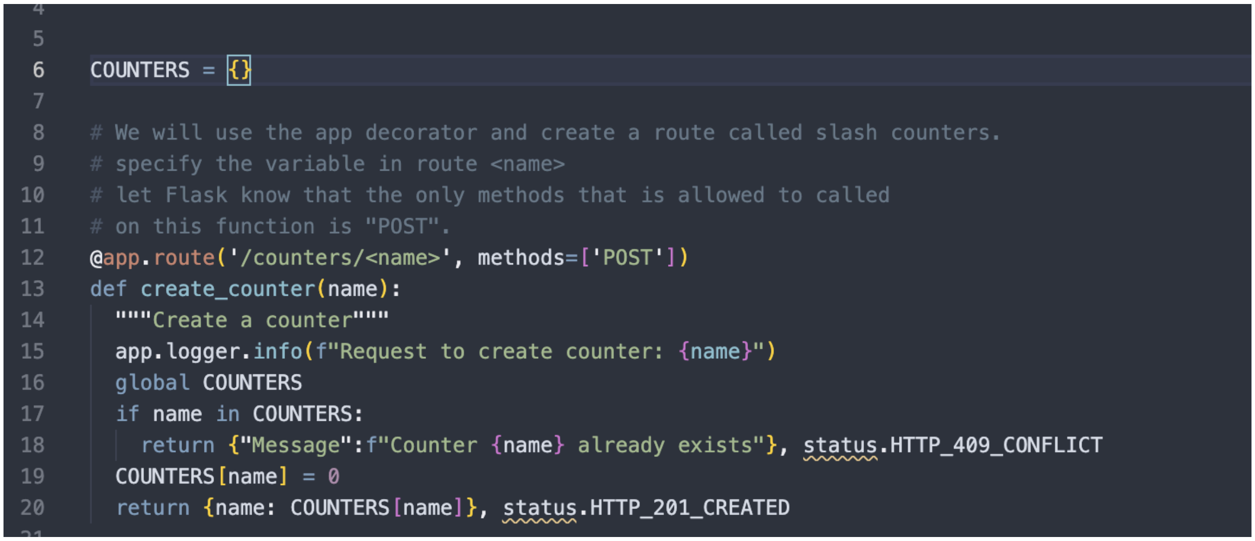
Task 5:

RED - AssertionError: 404 !=201

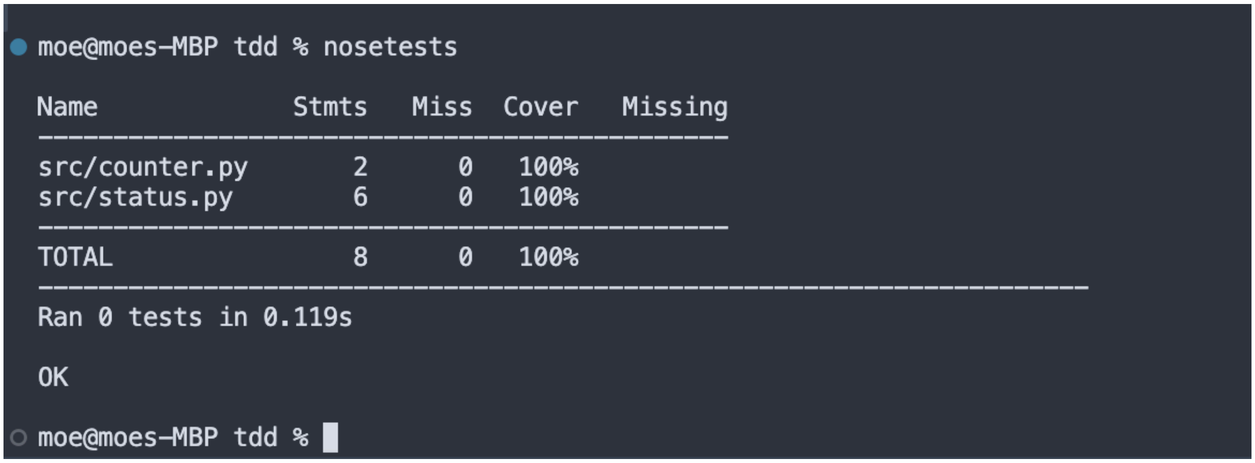


```
    from src.counter import app
ModuleNotFoundError: No module named 'src.counter'

Name            Stmts   Miss  Cover   Missing
---------------------------------------------
src/status.py       6      6     0%   2-7
---------------------------------------------
TOTAL               6      6     0%
----------------------------------------------------------------------
Ran 1 test in 0.005s

FAILED (errors=1)
```

Addition of code (def create_counter(name):)

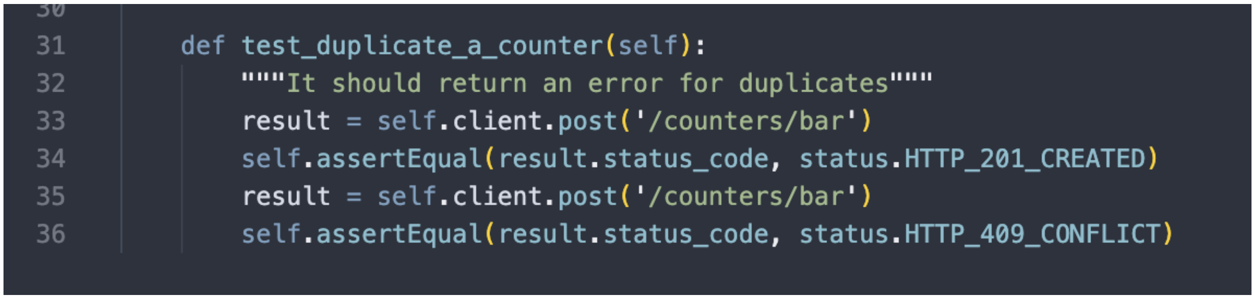


```

COUNTERS = {}

# We will use the app decorator and create a route called slash counters.
# specify the variable in route <name>
# let Flask know that the only methods that is allowed to called
# on this function is "POST".
@app.route('/counters/<name>', methods=['POST'])
def create_counter(name):
  """Create a counter"""
  app.logger.info(f"Request to create counter: {name}")
  global COUNTERS
  if name in COUNTERS:
    return {"Message":f"Counter {name} already exists"}, status.HTTP_409_CONFLICT
  COUNTERS[name] = 0
  return {name: COUNTERS[name]}, status.HTTP_201_CREATED
```



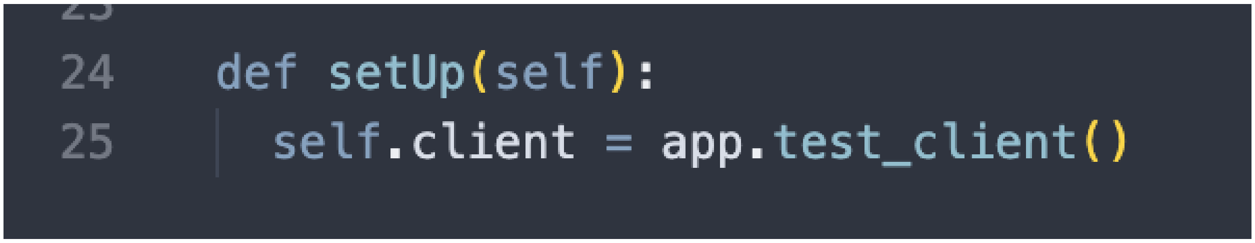
```
moe@moes-MBP tdd % nosetests

Name                Stmts   Miss  Cover   Missing
-------------------------------------------------
src/counter.py          2      0   100%
src/status.py           6      0   100%
-------------------------------------------------
TOTAL                   8      0   100%
----------------------------------------------------------------------
Ran 0 tests in 0.119s

OK

moe@moes-MBP tdd % 
```

Addition of code (def test_duplicate_a_counter(self):)



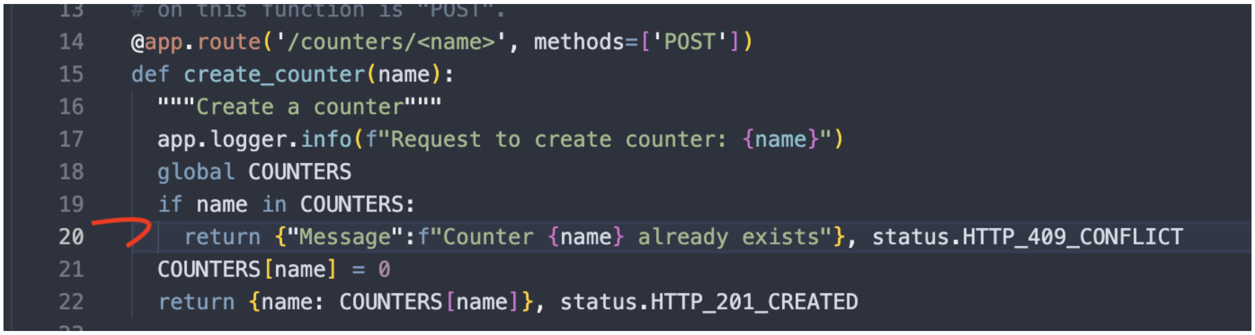
```python
    def test_duplicate_a_counter(self):
        """It should return an error for duplicates"""
        result = self.client.post('/counters/bar')
        self.assertEqual(result.status_code, status.HTTP_201_CREATED)
        result = self.client.post('/counters/bar')
        self.assertEqual(result.status_code, status.HTTP_409_CONFLICT)
```

Def setUp(self):



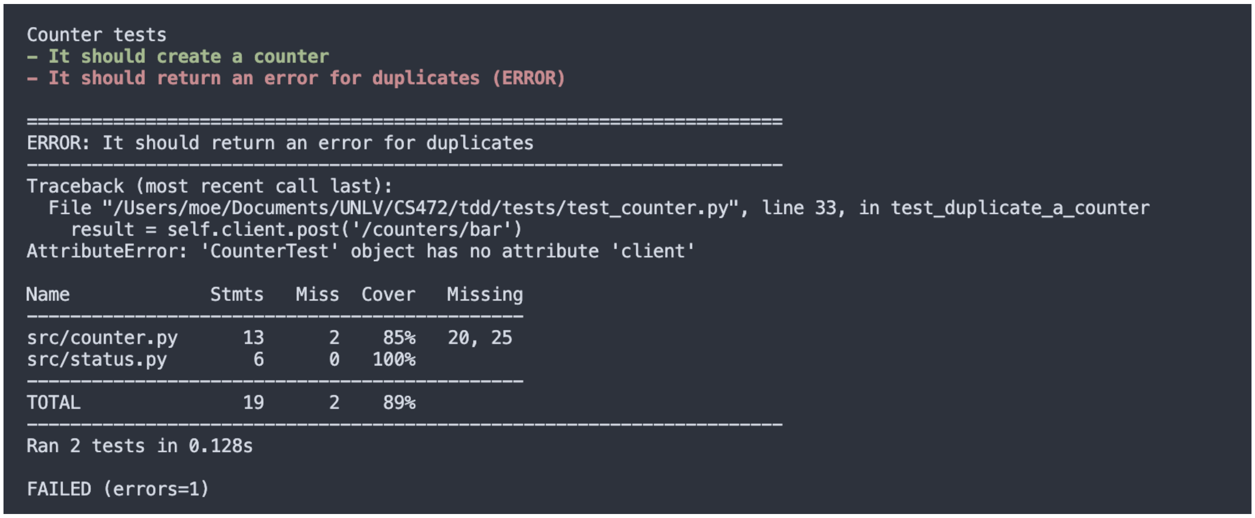
```python
    def setUp(self):
        self.client = app.test_client()
```

“If” statement added



```python
# on this function is "POST".
@app.route('/counters/<name>', methods=['POST'])
def create_counter(name):
    """Create a counter"""
    app.logger.info(f"Request to create counter: {name}")
    global COUNTERS
    if name in COUNTERS:
        return {"Message":f"Counter {name} already exists"}, status.HTTP_409_CONFLICT
    COUNTERS[name] = 0
    return {name: COUNTERS[name]}, status.HTTP_201_CREATED
```

Red and Green phases appear:



```
Counter tests
- It should create a counter
- It should return an error for duplicates (ERROR)

======================================================================
ERROR: It should return an error for duplicates
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/Users/moe/Documents/UNLV/CS472/tdd/tests/test_counter.py", line 33, in test_duplicate_a_counter
    result = self.client.post('/counters/bar')
AttributeError: 'CounterTest' object has no attribute 'client'

Name             Stmts   Miss  Cover   Missing
----------------------------------------------
src/counter.py      13      2    85%   20, 25
src/status.py        6      0   100%
----------------------------------------------
TOTAL               19      2    89%
----------------------------------------------------------------------
Ran 2 tests in 0.128s

FAILED (errors=1)
```