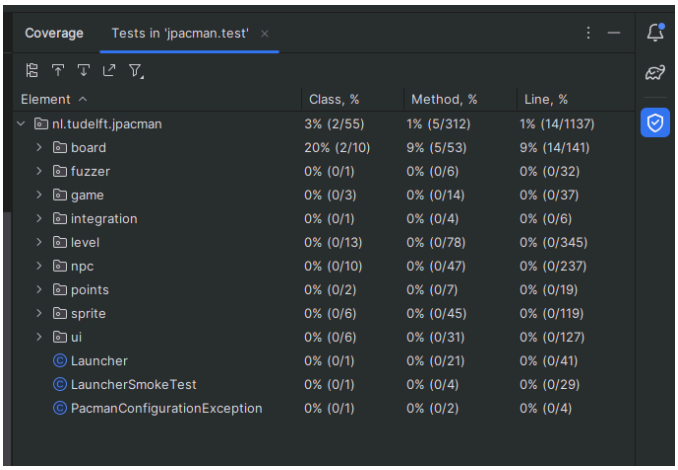
CS472 Lab 2

Fabrizio Valdivia

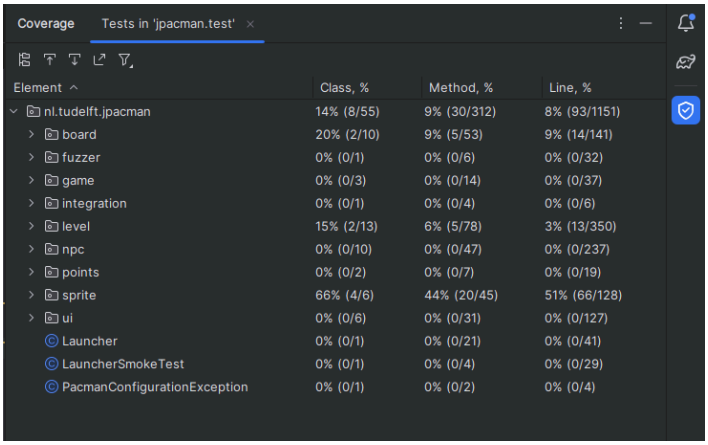
fork:https://github.com/Fabrizv/munch/tree/main

02/05/24

Task 1:

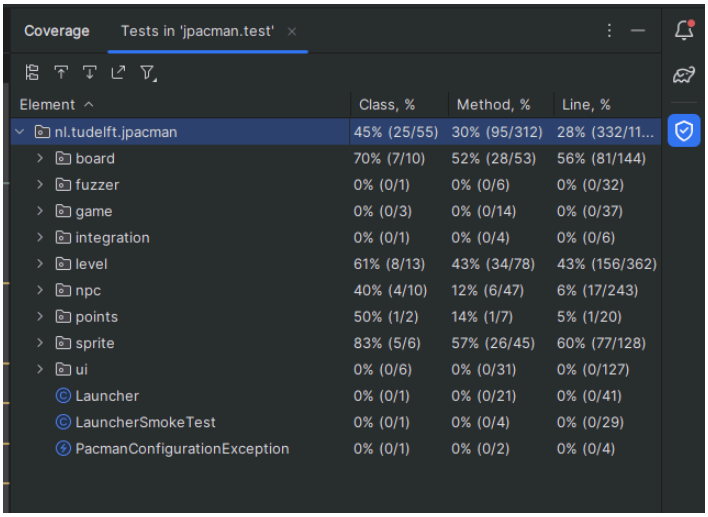
- After first running coverage without doing any tests
  - 

Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	3% (2/55)	1% (5/312)	1% (14/1137)
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	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)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

- After adding PlaterTest.java this is the 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)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

Task 2.1:

- After all tests are done:
  - 

Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	45% (25/55)	30% (95/312)	28% (332/11...
board	70% (7/10)	52% (28/53)	56% (81/144)
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	61% (8/13)	43% (34/78)	43% (156/362)
npc	40% (4/10)	12% (6/47)	6% (17/243)
points	50% (1/2)	14% (1/7)	5% (1/20)
sprite	83% (5/6)	57% (26/45)	60% (77/128)
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)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

CS472 Lab 2 Fabrizio Valdivia

fork:https://github.com/Fabrizv/munch/tree/main 02/05/24

- Here I increased the coverage for jPacman by working on some methods including, testParseMap(), SpriteTest(), and testPlayerVsGhost()
-
```java
public class mapParserTest {


    private static final PacManSprites SPRITE_STORE = new PacManSprites();

    private final BoardFactory BOARD_FACTORY = new BoardFactory(SPRITE_STORE);

    private final GhostFactory GHOST_FACTORY = new GhostFactory(SPRITE_STORE);

    private final LevelFactory LEVEL_FACTORY = new LevelFactory(SPRITE_STORE, GHOST_FACTORY, null);

    private final MapParser MAP_PARSER = new MapParser(LEVEL_FACTORY, BOARD_FACTORY);


    @Test
    void testParseMap() {
        List<String> testMap = Arrays.asList(
            "#####",
            "#...#",
            "#G P#",
            "#...#",
            "#####"
        );
        Level level = MAP_PARSER.parseMap(testMap);
        assertThat(level).isNotNull();
    }
}
```
    - In mapParserTest() I create the Pacman sprites, board/level/ghost factories, and the map parser. Then I make a test map to make sure it works
-
```java
public class SpriteTest {
    2 usages
    private EmptySprite emptySprite;
    new *
    @BeforeEach
    void setUp() {
        emptySprite = new EmptySprite();
    }
    new *
    @Test
    void testEmpty() {
        assertThat(emptySprite.getWidth()).isEqualTo( expected: 0);
    }
}
```
    - In SpriteTest() I make an empty sprite and make sure that is it in fact empty

```
public class playerVersusGhostTest {
    2 usages
    private static final PacManSprites SPRITES = new PacManSprites();
    1 usage
    private PlayerFactory playerFactory = new PlayerFactory(SPRITES);
    5 usages
    private Player player = playerFactory.createPacMan();
    1 usage
    private GhostFactory ghostFactory = new GhostFactory(SPRITES);
    2 usages
    private Ghost clyde = ghostFactory.createClyde();
    1 usage
    private PointCalculator pc = new DefaultPointCalculator();
    1 usage
    PlayerCollisions cp = new PlayerCollisions(pc);
    new *
    @Test
    void testPlayerVersusGhost(){
        int startScore = player.getScore();
        cp.playerVersusGhost(player, clyde);
        assertThat(player.isAlive()).isEqualTo( expected: false);
        assertThat(player.getKiller()).isEqualTo(clyde);
        assertThat( actual: player.getScore() == startScore);
    }
}
```

- In playerVersusGhostTest() I create a new PacmanSprite() and playerFactory() for the player, a ghostFactory() to create a ghost, and PlayerCollisions() with a DefaultPointCalculator(). Test if the player collides with a ghost, getting killed, then I get the score of the player and reset it.

Task 3:

jpacman Sessions

jpacman

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
nl.tudelft.jpacman.level		67%		57%	73	155	103	344	20	69	4	12
nl.tudelft.jpacman.npc.ghost		71%		55%	56	105	43	181	5	34	0	8
nl.tudelft.jpacman.ui		77%		47%	54	86	21	144	7	31	0	6
default		0%		0%	12	12	21	21	5	5	1	1
nl.tudelft.jpacman.board		86%		58%	44	93	2	110	0	40	0	7
nl.tudelft.jpacman.sprite		87%		59%	29	70	10	113	4	38	0	5
nl.tudelft.jpacman		69%		25%	12	30	18	52	6	24	1	2
nl.tudelft.jpacman.points		60%		75%	1	11	5	21	0	9	0	2
nl.tudelft.jpacman.game		87%		60%	10	24	4	45	2	14	0	3
nl.tudelft.jpacman.npc		100%		n/a	0	4	0	8	0	4	0	1
Total	1,208 of 4,694	74%	293 of 637	54%	291	590	227	1,039	49	268	6	47

Created with JaCoCo 0.8.3.201901230119

- The coverage results from JaCoCo is not similar to IntelliJ. This method shows it in a more visual format like with the bar graph. You can also click on each folder and file and see the statistics for each one
- I did not find this useful because there is no explanation of what JaCoCo means by the colors on the files. Also, I am colorblind and find it difficult to tell red and green apart…
- I prefer IntelliJ because it is simple to understand what it is saying and doesn't have so many numbers that make it overwhelming.

Task 4:

```
PS C:\Users\fabri\java projects\test\test_coverage> nosetests

Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data
- Test deleting an account
- Test creating from a dictionary
- Test the representation of an account
- Test account to dict
- Test updating an account
- Test updating an empty id

Name                  Stmts   Miss  Cover   Missing
-----------------------------------------------------
models\__init__.py        7      0   100%
models\account.py        40      0   100%
-----------------------------------------------------
TOTAL                    47      0   100%
----------------------------------------------------------------------
Ran 8 tests in 0.633s

OK

PS C:\Users\fabri\java projects\test\test_coverage>
```

- Here is my test_coverage after I got the coverage to 100%

```
new *
def test_from_dict(self):
    """ Test creating from a dictionary """
    data = ACCOUNT_DATA[self.rand]  # Get a random account
    account = Account(**data)
    result = Account()
    result.from_dict(account.to_dict())
    self.assertEqual(account.name, result.name)
    self.assertEqual(account.email, result.email)
    self.assertEqual(account.phone_number, result.phone_number)
    self.assertEqual(account.disabled, result.disabled)
    self.assertEqual(account.date_joined, result.date_joined)
```

- This is my test_from_dict() function. I get a random account data and save it as an account. I make another account and use the from_dict() function to turn it to a dictionary. Then I check that all the data is equal.

```python
def test_update(self):
        """ Test updating an account """
        import datetime
        data = ACCOUNT_DATA[self.rand]  # get a random account
        account = Account(**data)
        account.create()
        account.name = "Updated Name"
        account.email= "updated@example.com"
        account.phone_number = 5555555555
        account.disabled = 0
        account.date_joined = datetime.datetime(1000,1,1)
        account.update()
        updated_account = Account.find(account.id)
        self.assertEqual(updated_account.name, "Updated Name")
        self.assertEqual(account.email, "updated@example.com")
        self.assertEqual(account.phone_number, "5555555555")
        self.assertEqual(account.disabled, False)
        self.assertEqual(account.date_joined, datetime.datetime(1000,1,1))
```

-
  - This is my test_update() function. Here I get a random account and then I use .create() and set the name, email, phone number, disabled, and date joined. The I use .update() to update everything. Then I check to make sure all the categories were updated correctly.

```python
def test_update_empty_id(self):
        """ Test updating an empty id """
        data = ACCOUNT_DATA[self.rand]  # get a random account
        account = Account(**data)
        with self.assertRaises(DataValidationError):
            account.update()

def test_delete(self):
    """ Test deleting an account """
    data = ACCOUNT_DATA[self.rand]  # get a random account
    account = Account(**data)
    account.create()
    id = account.id
    account.delete()
    self.assertIsNone(Account.find(id))
```

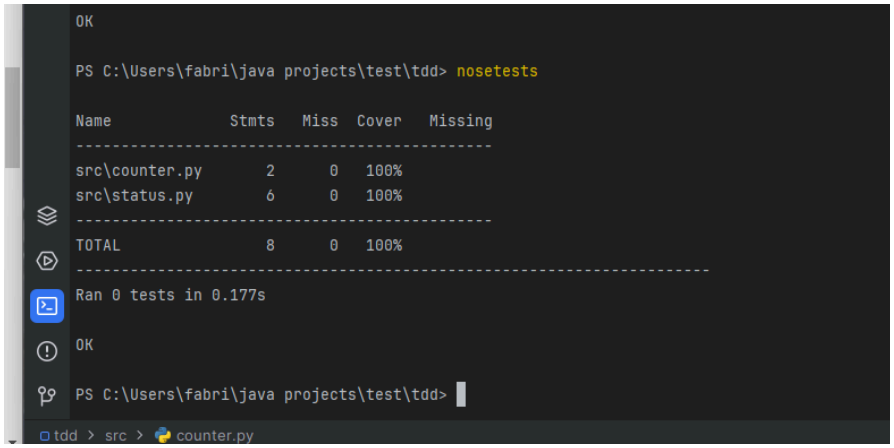
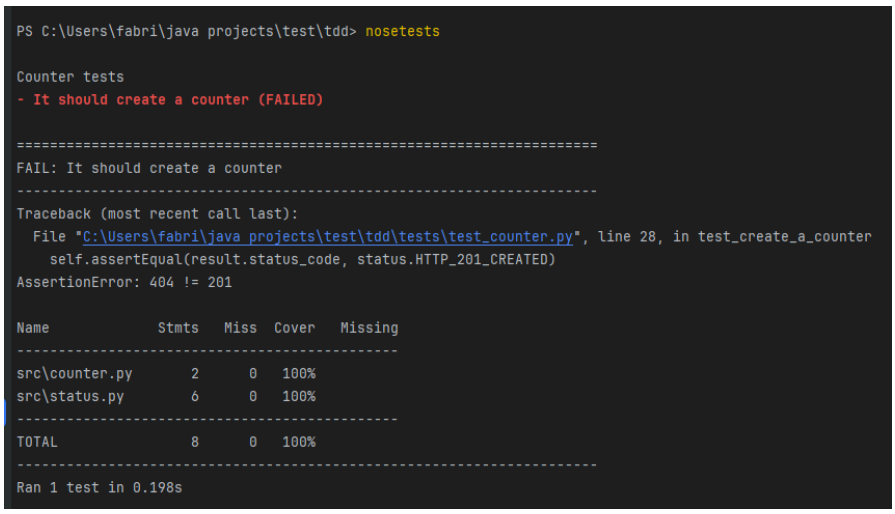
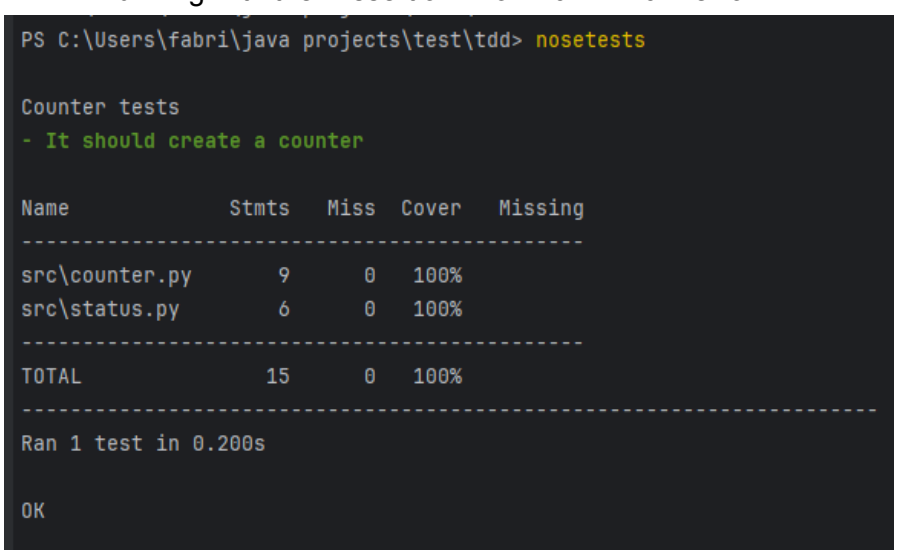
-
  - These are the test_update_empty_id() and test_delete() functions.
  - In the test_update_empty_id() function I get a random account and chekc if there is a DataValidationError, if so then update the account.
  - In the test_delete() function I get a random account, save the ID associated to it and then delete the account. After that I make sure that the account can't be found by using the ID to look for it.

Task 5:

CS472 Lab 2

fork:https://github.com/Fabrizv/munch/tree/main

Fabrizio Valdivia
02/05/24

- 
```
OK

PS C:\Users\fabri\java projects\test\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 0.177s

OK

PS C:\Users\fabri\java projects\test\tdd>
```
  - Running without the test cases
- 
```
PS C:\Users\fabri\java projects\test\tdd> nosetests

Counter tests
- It should create a counter (FAILED)

======================================================================
FAIL: It should create a counter
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\fabri\java projects\test\tdd\tests\test_counter.py", line 28, in test_create_a_counter
    self.assertEqual(result.status_code, status.HTTP_201_CREATED)
AssertionError: 404 != 201

Name              Stmts   Miss  Cover   Missing
-----------------------------------------------
src\counter.py        2      0   100%
src\status.py         6      0   100%
-----------------------------------------------
TOTAL                 8      0   100%
----------------------------------------------------------------------
Ran 1 test in 0.198s
```
  - Running with the “AssertionError: 404 != 201” error
- 
```
PS C:\Users\fabri\java projects\test\tdd> nosetests

Counter tests
- It should create a counter

Name              Stmts   Miss  Cover   Missing
-----------------------------------------------
src\counter.py        9      0   100%
src\status.py         6      0   100%
-----------------------------------------------
TOTAL                15      0   100%
----------------------------------------------------------------------
Ran 1 test in 0.200s

OK
```
  - Running with the previous error resolved

CS472 Lab 2

Fabrizio Valdivia

fork:https://github.com/Fabrizv/munch/tree/main

02/05/24

- ```
  PS C:\Users\fabri\java projects\test\tdd> nosetests

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

  ======================================================================
  FAIL: It should return an error for duplicates
  ----------------------------------------------------------------------
  Traceback (most recent call last):
    File "C:\Users\fabri\java projects\test\tdd\tests\test_counter.py", line 38, in test_duplicate_a_counter
      self.assertEqual(result.status_code, status.HTTP_409_CONFLICT)
  AssertionError: 201 != 409
  -------------------- >> begin captured logging << --------------------
  src.counter: INFO: Request to create counter: bar
  src.counter: INFO: Request to create counter: bar
  -------------------- >> end captured logging << --------------------

  Name             Stmts   Miss  Cover   Missing
  ---------------------------------------------
  ```
  - Running with the “AssertionError: 201 != 409” error
- ```
  PS C:\Users\fabri\java projects\test\tdd> nosetests

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

  Name             Stmts   Miss  Cover   Missing
  ---------------------------------------------
  src\counter.py      11      0   100%
  src\status.py        6      0   100%
  ---------------------------------------------
  TOTAL               17      0   100%
  ----------------------------------------------------------------------
  Ran 2 tests in 0.208s

  OK
  ```
  - Running with the previous error fixed

CS472 Lab 2 Fabrizio Valdivia

fork:https://github.com/Fabrizv/munch/tree/main 02/05/24

```python
def test_update_a_counter(self):
    """It should update a counter"""
    result = self.client.post('/counters/uc')
    self.assertEqual(result.status_code, status.HTTP_201_CREATED)

    base_result = self.client.get('/counters/uc')
    base_value = json.loads(base_result.data)["uc"]

    update_result = self.client.put('/counters/uc')
    self.assertEqual(update_result.status_code, status.HTTP_200_OK)

    new_result = self.client.get('/counters/uc')
    new_value = json.loads(new_result.data)["uc"]
    self.assertEqual(new_value, base_value + 1)

    no_result = self.client.put('/counters/no_uc')
    self.assertEqual(no_result.status_code, status.HTTP_404_NOT_FOUND)
    response_data = json.loads(no_result.data)
    self.assertEqual(response_data["error"], second: "Counter not found")
```

-

```
PS C:\Users\fabri\java projects\test\tdd> nosetests

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

======================================================================
ERROR: It should update a counter
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\fabri\java projects\test\tdd\tests\test_counter.py", line 47, in test_update_a_counter
    base_value = json.loads(base_result.data)["bar_update"]
  File "c:\users\fabri\appdata\local\programs\python\python39-32\lib\json\__init__.py", line 346, in loads
    return _default_decoder.decode(s)
  File "c:\users\fabri\appdata\local\programs\python\python39-32\lib\json\decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "c:\users\fabri\appdata\local\programs\python\python39-32\lib\json\decoder.py", line 355, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
-------------------- >> begin captured logging << --------------------
src.counter: INFO: Request to create counter: bar_update
-------------------- >> end captured logging << --------------------

Name             Stmts   Miss  Cover   Missing
-----------------------------------------------
```

  - In the code, I make a POST request and check that it has the correct status code. Then I make a GET and PUT request and check that the updated is the same with the status code. Later I do a GET request and check that the value of this one is one more than the first GET. Then I make PUT request and check to make sure it doesn't exist and give the correct status code. Finally it would display the "counter not found message".
  - Running with the update counter erroring

CS472 Lab 2

Fabrizio Valdivia

fork:https://github.com/Fabrizv/munch/tree/main

02/05/24

- ```python
  @app.route( rule: '/counters/<name>', methods=['PUT'])
  def update_counter(name):
      """Update a counter"""
      global COUNTERS
      if name not in COUNTERS:
          return {"error": "Counter not found"}, status.HTTP_404_NOT_FOUND
      COUNTERS[name] += 1
      return {name: COUNTERS[name]}, status.HTTP_200_OK
  ```
- ```
  PS C:\Users\fabri\java projects\test\tdd> nosetests

  Counter tests
  - It should create a counter
  - It should return an error for duplicates
  - It should update a counter

  Name              Stmts   Miss  Cover   Missing
  ----------------------------------------------
  src\counter.py       17      1    94%   26
  src\status.py         6      0   100%
  ----------------------------------------------
  TOTAL                23      1    96%
  ----------------------------------------------------------------------
  Ran 3 tests in 0.216s

  OK
  ```
  - In the code I check if name is in COUNTERS, if not then it returns an error, or else it increments its value by 1 and returns.
  - Running with the previous error fixed
- ```python
  new *
  def test_read_counter(self):
      """It should read a counter"""
      client = app.test_client()
      result = client.post('/counters/rc')
      result = client.get('/counters/rc')
      self.assertEqual(result.status_code, status.HTTP_200_OK)
      result2 = client.get('/counters/doesnotexist')
      self.assertEqual(result2.status_code, status.HTTP_404_NOT_FOUND)
  ```
  ```

CS472 Lab 2

Fabrizio Valdivia

fork:https://github.com/Fabrizv/munch/tree/main

02/05/24

- ```
  PS C:\Users\fabri\java projects\test\tdd> nosetests

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

  ======================================================================
  ERROR: It should update a counter
  ----------------------------------------------------------------------
  Traceback (most recent call last):
    File "C:\Users\fabri\java projects\test\tdd\tests\test_counter.py", line 44, in test_update_a_counter
      base_value = json.loads(base_result.data)["bar_update"]
    File "c:\users\fabri\appdata\local\programs\python\python39-32\lib\json__init__.py", line 346, in loads
      return _default_decoder.decode(s)
    File "c:\users\fabri\appdata\local\programs\python\python39-32\lib\json\decoder.py", line 337, in decode
      obj, end = self.raw_decode(s, idx=_w(s, 0).end())
    File "c:\users\fabri\appdata\local\programs\python\python39-32\lib\json\decoder.py", line 355, in raw_decode
      raise JSONDecodeError("Expecting value", s, err.value) from None
  json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
  -------------------- >> begin captured logging << --------------------
  src.counter: INFO: Request to create counter: bar_update
  -------------------- >> end captured logging << --------------------

  ======================================================================
  FAIL: It should read a counter
  ----------------------------------------------------------------------
  Traceback (most recent call last):
    File "C:\Users\fabri\java projects\test\tdd\tests\test_counter.py", line 64, in test_read_counter
  ```
  - In the code I make a client and make a POST and GET request. Then I check if the status code exists and is good. Finally, I do a GET request and check if the status code is 404 not found.
  - Running with the read counter error
- ```
  @app.route('/counters/<name>', methods=['GET'])
  def read_counter(name):
      global COUNTERS
      if name not in COUNTERS:
          return {"error": "Counter not found"}, status.HTTP_404_NOT_FOUND
      return {name: COUNTERS[name]}, status.HTTP_200_OK
  ```

```
PS C:\Users\fabri\java projects\test\tdd> nosetests

Counter tests
- It should create a counter
- It should return an error for duplicates
- It should read a counter
- It should update a counter

Name              Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py       22      0   100%
src\status.py         6      0   100%
----------------------------------------------
TOTAL                28      0   100%
----------------------------------------------------------------------
Ran 4 tests in 0.312s

OK

PS C:\Users\fabri\java projects\test\tdd>
```

-
    - In the code, I check if name is in COUNTERS, if not then it returns an error, else it returns correctly
    - Running with the previous error fixed