과제

First Sprint

장유선

2023.09.08

1. 문제 정의

설계한 내용을 TDD 방식으로 구현

로그인

관리자 계정: admin / admin 사용자 계정: user01 / password 위 2개 계정 이외에는 로그인 실패 로그인 시 관리자는 admin, 사용자 계정은 user를 반환

POI 편집

단일 POI GET 요청 시 아래의 데이터 리스트 반환 ID: 1 NAME: 테스트용 이외의 ID가 입력 되는 경우는 None 반환

POI 추가 시

Name이 None이 아닌 경우 ID는 2를 반환 Name이 None인 경우 ID는 None 반환

POI 편집 시

ID가 1이고, NAME이 None이 아닌 경우 True 반환 이외의 경우에는 False 반환

POI 삭제 시

ID가 1인 경우 True 반환 이외의 경우에는 False 반환

2. POI.py

```
class POI:
    def __init__(self):
        self.poi_list = {
            1: "테스트용",
        }

    def login(self, username, password):
        if username == "admin" and password == "admin":
            return "admin"
        elif username == "user01" and password == "password":
            return "user"
        else:
            return None

    def get_poi(self, poi_id):
        return self.poi_list.get(poi_id)

    def add_poi(self, name):
        if name is not None:
            new_id = 2
            return new_id
        else:
            return None

    def edit_poi(self, poi_id, name):
        if poi_id == 1 and name is not None:
            return True
        else:
            return False

    def delete_poi(self, poi_id):
        if poi_id == 1:
            return True
        else:
            return False
```

3. test_POI.py

```
import unittest
from unittest import TestCase
from POI import *

class TestPOI(TestCase):
    def setUp(self):
        self.poi = POI()

    def test_login_admin(self):
        self.assertEqual(self.poi.login("admin", "admin"), "admin")

    def test_login_user(self):
        self.assertEqual(self.poi.login("user01", "password"), "user")

    def test_login_fail(self):
        self.assertIsNone(self.poi.login("me", "me"))

    def test_get_poi_existing(self):
        self.assertEqual(self.poi.get_poi(1), "테스트용")

    def test_get_poi_non_existing(self):
        self.assertIsNone(self.poi.get_poi(2))

    def test_add_poi_with_name(self):
        self.assertEqual(self.poi.add_poi("마라탕집"), 2)

    def test_add_poi_without_name(self):
        self.assertIsNone(self.poi.add_poi(None))

    def test_edit_poi_valid(self):
        self.assertTrue(self.poi.edit_poi(1, "POI 수정"))

    def test_edit_poi_invalid(self):
        self.assertFalse(self.poi.edit_poi(2, "POI 수정"))

    def test_delete_poi_valid(self):
        self.assertTrue(self.poi.delete_poi(1))

    def test_delete_poi_invalid(self):
        self.assertFalse(self.poi.delete_poi(2))

if __name__ == "__main__":
    unittest.main()
```

3. 결과

Total number of tests expected to run: 11

Total number of tests run: 11

Total number of tests passed: 11

Total number of tests failed: 0

Total number of tests failed with errors: 0

Total number of tests skipped: 0

Finished running tests!

4. 결과 화면