Module 450 – Evaluation pratique finale

Lien vers Repository GitHub :

https://github.com/gchopard2/450_web_test

Lien vers le PDF sur GitHub :

https://github.com/gchopard2/450_web_test/blob/examen/docs/450.pdf

Les plans de tests

Dans ce même document, [Accéder ici]

Rapport de tests simplifié

Dans ce même document, [Accéder ici]

Un pipeline sur GitHub a également été mis en place :

Le fichier de pipeline :
https://github.com/gchopard2/450_web_test/blob/examen/.github/workflows/python-app.yml

L'historique des exécutions :

https://github.com/gchopard2/450_web_test/actions

Utilisation d'un logiciel de contrôle statique du code

Dans ce même document, [Accéder ici]

Les plans de tests

Plan de test 1 : Le modèle DVD – Test de l'objet

Introduction et objectifs

Le plan de test présent ici va permettre de tester la nouvelle fonctionnalité, permettant de sauvegarder des DVD dans l'application. Le but est de vérifier si les données sont correctement sauvegardées.

Étapes du test

2 étapes simples :

- Créer un DVD
- Vérifier que l'objet converti en texte retourne les données prévues.

Résultat Attendu :

Il est attendu que :

- La création du DVD fonctionne correctement
- L'objet converti en texte retournent les données prévues

La mise en place :

https://github.com/gchopard2/450_web_test/blob/examen/library/tests/test_model_dvd.py

```python
class DVDModelTest(TestCase):

    @classmethod
    def setUpTestData(cls):
        # Création du DVD
        DVD.objects.create(title = "Pico Bogue", duration = 120, pegi = "16+", description = "Pico Bogue - a great movie")

    def test_dvd_model_str(self):
        # Obtenir le DVD de la BD
        dvd = DVD.objects.get(id=1)

        # Valeur attendue
        expected_value = "Pico Bogue | 120mn | 16+ | Pico Bogue - a great movie"

        # Tester
        assert str(dvd) == expected_value
```

Le résultat :

```
library/tests/test_model_dvd.py::DVDModelTest::test_dvd_model_str PASSED
```

La création du DVD à réussi, et l'objet converti en texte correspond bien à ce qui est demandé.

Plan de test 2 : Le modèle DVD – Test des attributs

Introduction et objectifs

Le but de ce 2ème test est de vérifier si les attributs de l'objet sont correctement sauvegardés

Étapes du test

2 étapes simples :

- Créer un DVD
- Vérifier chaque attribut de l'objet pour qu'ils soient égaux aux valeurs sauvegardées.

Résultat Attendu :

Il est attendu que :

- La création du DVD fonctionne correctement
- Les attributs de l'objet soie égal à ce qui a été ajouté dans la base de données.

La mise en place :

https://github.com/gchopard2/450_web_test/blob/examen/library/tests/test_model_dvd.py

```python
@classmethod
def setUpTestData(cls):
    # Création du DVD
    DVD.objects.create(title = "Pico Bogue", duration = 120, pegi = "16+", description = "Pico Bogue - a great movie")
```

```python
def test_dvd_model_content(self):
    # Obtenir le DVD de la BD
    dvd = DVD.objects.get(id=1)

    # Valeurs attendue
    title_field = "Pico Bogue"
    duration_field = 120
    pegi_field = "16+"
    description_field = "Pico Bogue - a great movie"

    # Tester
    assert title_field == dvd.title
    assert duration_field == dvd.duration
    assert pegi_field == dvd.pegi
    assert description_field == dvd.description
```

Le résultat :

```
library/tests/test_model_dvd.py::DVDModelTest::test_dvd_model_content PASSED
```

L'objet a été crée correctement, et les attributs correspondent bien à ce qui a été créé.

Plan de test 3 : Le nombre de DVD dans la base de données

Le but de ce 3ème test est de vérifier si sauvegarder plusieurs DVD fonctionne correctement.

Étapes du test

4 étapes simples :

- Créer 10 DVD avec des attributs incrémentiel
- Vérifier que les objets, converti en texte, retourne les données prévues.
- Pour chaque objet également, vérifier chaque attribut de l'objet pour qu'ils soient égaux aux valeurs sauvegardées.
- Et finalement, Vérifier le nombre d'objet dans la base de données, pour qu'ils correspondent bien à ce qui a été demandé.

Résultat Attendu :

Il est attendu que :

- La création des objets fonctionne correctement
- Les attributs des objets soie égal à ce qui a été ajouté dans la base de données.
- Que les objets convertis en texte retournent bien les données prévues.
- Que le nombre d'objets sauvegardé corresponde.

La mise en place :

https://github.com/gchopard2/450_web_test/blob/examen/library/tests/test_dvd_count.py

```python
import pytest

from django.test import Client, TestCase
from library.models import DVD

AmountToCreate = 10

class DVDModelTest(TestCase):
    @classmethod
    def setUpTestData(cls):
        for i in range(AmountToCreate):
            DVD.objects.create(title="Pico Bogue "+str(i), duration=120, pegi="16+", description="Pico Bogue - a great movie "+str(i))

    def test_dvd_model_str(self):
        for i in range(AmountToCreate):
            dvd = DVD.objects.get(id=i+1)
            expected_value = "Pico Bogue "+str(i)+" | 120mn | 16+ | Pico Bogue - a great movie "+str(i)

            assert str(dvd) == expected_value

    def test_dvd_model_content(self):
        for i in range(AmountToCreate):
            dvd = DVD.objects.get(id=i+1)
            title_field = dvd.title
            duration_field = dvd.duration
            pegi_field = dvd.pegi
            description_field = dvd.description

            assert title_field == "Pico Bogue "+str(i)
            assert duration_field == 120
            assert pegi_field == "16+"
            assert description_field == "Pico Bogue - a great movie "+str(i)

    def test_dvd_count(self):
        assert DVD.objects.count() == AmountToCreate
```

Le résultat :

```
library/tests/test_dvd_count.py::DVDModelTest::test_dvd_count PASSED
library/tests/test_dvd_count.py::DVDModelTest::test_dvd_model_content PASSED
library/tests/test_dvd_count.py::DVDModelTest::test_dvd_model_str PASSED
```

Les DVDs sont créés correctement, les attributs ainsi que les objets convertis en texte correspondent bien à ce qui est demandé, le nombre de dvd inséré correspond bien également.

Rapport de tests simplifié

Ici se trouve le résultat des tests :

```
PS C:\Data\MODULES\450\450_web_test> pytest -v
========================================= test session starts =========================================
platform win32 -- Python 3.12.0, pytest-7.4.3, pluggy-1.3.0 -- C:\Program Files\Python312\python.exe
cachedir: .pytest_cache
django: version: 5.0.3, settings: epsic_library.settings (from ini)
rootdir: C:\Data\MODULES\450\450_web_test
configfile: pytest.ini
plugins: cov-4.1.0, django-4.8.0, mock-3.12.0
collected 10 items

library/tests/test_book_count.py::BookModelTest::test_book_count PASSED                  [ 10%]
library/tests/test_book_count.py::BookModelTest::test_book_model_content PASSED          [ 20%]
library/tests/test_book_count.py::BookModelTest::test_book_model_str PASSED              [ 30%]
library/tests/test_dvd_count.py::DVDModelTest::test_dvd_count PASSED                     [ 40%]
library/tests/test_dvd_count.py::DVDModelTest::test_dvd_model_content PASSED             [ 50%]
library/tests/test_dvd_count.py::DVDModelTest::test_dvd_model_str PASSED                 [ 60%]
library/tests/test_model_book.py::BookModelTest::test_book_model_content PASSED          [ 70%]
library/tests/test_model_book.py::BookModelTest::test_book_model_str PASSED              [ 80%]
library/tests/test_model_dvd.py::DVDModelTest::test_dvd_model_content PASSED             [ 90%]
library/tests/test_model_dvd.py::DVDModelTest::test_dvd_model_str PASSED                 [100%]

========================================= 10 passed in 0.32s ==========================================
PS C:\Data\MODULES\450\450_web_test>
```

Si on se concentre sur l'aspect des tests de DVDs, tous les tests passent.

Les résultats pour chaque test :

- library/tests/test_dvd_count.py::DVDModelTest::test_dvd_count
  - Celui-ci teste le nombre de DVD crée dans la base de données, et il a réussi.
- library/tests/test_dvd_count.py::DVDModelTest::test_dvd_model_content
  - Celui-ci teste chaque dvd pour ses attributs, tester si l'attribut sauvegardé corresponde bien au DVD sauvegardé. Le test a réussi.
- library/tests/test_dvd_count.py::DVDModelTest::test_dvd_model_str
  - Ce test vérifie si les DVD converti depuis un objet en texte corresponde bien à ce qui a été ajouté à la base de données. Le test a réussi.
- library/tests/test_model_dvd.py::DVDModelTest::test_dvd_model_content
  - Ce test vérifie pour un DVD, si les attributs correspondent bien à ce qui a été ajouté. Le test a réussi.
- library/tests/test_model_dvd.py::DVDModelTest::test_dvd_model_str
  - Ce test vérifie sur 1 DVD, si l'objet converti en texte correspond bien à ce qui a été ajouté à la base de données. Le test a réussi.

Utilisation d'un logiciel de contrôle statique du code

A l'aide de Flake8, il est possible de faire tester le code pour en assurer une bonne qualité

```
PS C:\Data\MODULES\450\450_web_test> flake8 --statistics .\library\ .\epsic_library\ --extend-exclude test_*.py
.\epsic_library\settings.py:23:80: E501 line too long (81 > 79 characters)
.\epsic_library\settings.py:40:15: E261 at least two spaces before inline comment
.\epsic_library\settings.py:90:80: E501 line too long (91 > 79 characters)
.\epsic_library\settings.py:93:80: E501 line too long (81 > 79 characters)
.\epsic_library\settings.py:96:80: E501 line too long (82 > 79 characters)
.\epsic_library\settings.py:99:80: E501 line too long (83 > 79 characters)
.\epsic_library\urls.py:24:25: E231 missing whitespace after ','
.\epsic_library\urls.py:25:24: E231 missing whitespace after ','
.\library\admin.py:7:25: W292 no newline at end of file
.\library\migrations\0001_initial.py:17:80: E501 line too long (117 > 79 characters)
.\library\migrations\0002_create_dvd_table.py:5:1: E302 expected 2 blank lines, found 1
.\library\migrations\0002_create_dvd_table.py:14:80: E501 line too long (117 > 79 characters)
.\library\models.py:3:1: F401 'django.utils.timezone' imported but unused
.\library\models.py:5:1: E302 expected 2 blank lines, found 1
.\library\models.py:12:1: E302 expected 2 blank lines, found 1
.\library\models.py:19:80: E501 line too long (85 > 79 characters)
.\library\models.py:19:86: W292 no newline at end of file
.\library\views.py:5:1: E302 expected 2 blank lines, found 1
.\library\views.py:15:80: E501 line too long (80 > 79 characters)
.\library\views.py:22:9: E124 closing bracket does not match visual indentation
.\library\views.py:25:1: E302 expected 2 blank lines, found 1
.\library\views.py:28:53: E231 missing whitespace after ':'
.\library\views.py:30:1: E302 expected 2 blank lines, found 1
.\library\views.py:33:51: E231 missing whitespace after ':'
.\library\views.py:33:57: W292 no newline at end of file
1     E124 closing bracket does not match visual indentation
4     E231 missing whitespace after ','
1     E261 at least two spaces before inline comment
6     E302 expected 2 blank lines, found 1
9     E501 line too long (81 > 79 characters)
1     F401 'django.utils.timezone' imported but unused
3     W292 no newline at end of file
PS C:\Data\MODULES\450\450_web_test>
```

Le résultat présenté ici peut se résumer de cette manière :

- Certaines lignes sont trop longues
- Il y a des espaces en trop à la fin de certaines lignes
- Parfois, pas de retour à la ligne en fin de fichier
- Des retours à la ligne qui manque
- Des imports effectués mais jamais utilisé.