

Pruebas unitarias

Introducción

Este documento muestra y explica la realización de las pruebas unitarias y su resultado.

Estas pruebas unitarias son prácticas para verificar el comportamiento de la aplicación, como este proyecto esta realizado en Django las pruebas se realizaron con el modulo `unittest` de python.

Tests

Para la realización de los tests se crea una carpeta llamada tests dentro de la carpeta de prestamos (carpeta de la "aplicación" en Django) y allí se crean los distintos archivos para realizar las pruebas

Tests unitarios de URLs

Para realizar las pruebas unitarias de las URLs se creo el archivo test_urls.py y en este se importaron los siguientes módulos

```
1 from django.conf.urls import url
2 from django.test import TestCase
3 from django.urls import reverse, resolve
4 from prestamos.views import index, log_admin, administration, my_loans, loans,
```

Los cuales permitirán hacer las pruebas para cada pagina, se crea un TestCase dentro del cual se harán todas las pruebas.

A continuación se muestra una de las pruebas como ejemplo:

```
1 class Test_all_Urls (TestCase):
2
3     def test_url_index(self):
4         url = reverse ('index')
5         self.assertEqual(resolve(url).func, index)
```

En este caso la prueba se realiza con la pagina index, y lo que busca la prueba es comprobar que la pagina tenga asignada la URL correcta.

Se realiza el mismo procedimiento con los demás paginas.

Ejecución de las pruebas

Para ejecutar las pruebas unitarias en Django solo se debe estar dentro de la carpeta del proyecto desde la consola de comandos y allí escribir la siguiente línea de comando

```
python manage.py test prestamos.tests.test_urls
```

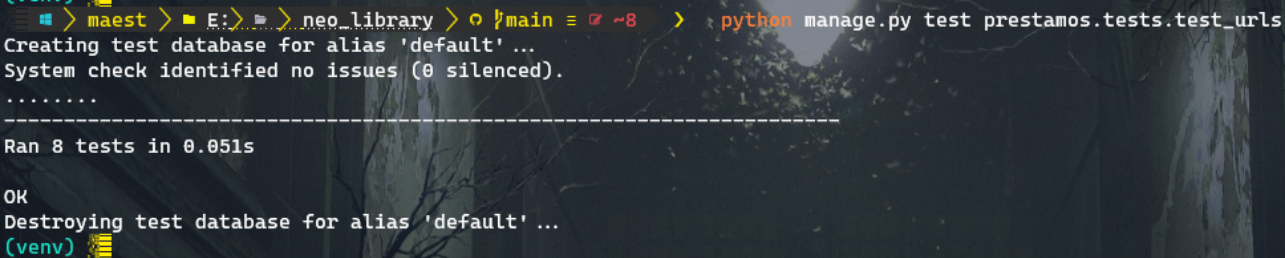
Que realizara todas las pruebas en el archivo test_urls el cual devuelve un resultado general de las pruebas, si todas tienen un resultado exitoso regresara un ok e indicará el número de pruebas realizadas y si alguna falla indica cuál de ellas falló.

Y si se desea realizar una prueba única de una URL específica se usa el siguiente código

```
python manage.py test prestamos.tests.test_urls.Test_all_urls.test_url_index
```

que corre una prueba sobre solo una página, en este caso index y retornará un ok si es correcta o si es un error retorna el mensaje de fallo.

Resultados de las pruebas



```
> maest > E:\neo_library > main > python manage.py test prestamos.tests.test_urls
Creating test database for alias 'default' ...
System check identified no issues (0 silenced).
.....
-----
Ran 8 tests in 0.051s
OK
Destroying test database for alias 'default' ...
(venv)
```

Se realizaron ocho pruebas y todas tuvieron un resultado exitoso

Test unitarios de views

Para realizar las pruebas unitarias de las views se creó el archivo test_views.py y en este se importaron los siguientes módulos

```
1 from django.test import TestCase
2 from django.urls import reverse
3 from django.contrib.auth.models import User
```

Los cuales permitirán hacer las pruebas para comprobar que las paginas respondan correctamente cuando las llamen y pruebas para comprobar si el usuario puede iniciar sesión.

Al igual que en las pruebas anteriores se crean dentro de un TestCase

El siguiente un ejemplo de las pruebas de respuesta de la pagina

```
1 class testviews (TestCase):
2
3     def test_index_view_correct (self):
4         resp = self.client.get(reverse('index'))
5         self.assertEqual(resp.status_code, 200)
6
7         self.assertTemplateUsed(resp, 'index.html')
```

Y el siguiente de es un ejemplo de las pruebas para comprobar que el usuario inicie sesión correctamente

```
1 class testviews (TestCase):
2
3     def setUp(self):
4         #Create one users
5         test_user1 = User.objects.create_user(username='testuser1', password='12345')
6         test_user1.save()
7
8
9
10    def test_logged_in_uses_correct_template_log_Admin(self):
11        login = self.client.login(username='testuser1', password='12345')
12        resp = self.client.get(reverse('log_admin'))
13
14        #Check our user is logged in
15        self.assertEqual(str(resp.context['user']), 'testuser1')
16        #Check that we got a response "success"
```

```
17         self.assertEqual(resp.status_code, 200)
18
19         #Check we used correct template
20         self.assertTemplateUsed(resp, 'loginad.html')
```

Para comprobar el inicio de sesión se crea un usuario temporal y se guarda momentáneamente y al mismo tiempo también se comprueba que la pagina responda correctamente

Las pruebas para inicio de sesión solo se realiza para las paginas que tienen inicio de sesión, pero la prueba para ver que las paginas respondan correctamente se realiza en todas.

Ejecución de las pruebas

Para ejecutar las pruebas unitarias en Django solo se debe estar dentro de la carpeta del proyecto desde la consola de comandos y allí escribir la siguiente línea de comando

```
python manage.py test prestamos.tests.test_views
```

Que realizara todas las pruebas en el archivo test_views el cual de vuelve un resultado general de las pruebas, si todas tienen un resultado exitoso regresara un ok e indicar el numero de pruebas realizadas y si alguna falla indica cual de ellas fallo.

Y si se desea realizar una prueba única de una view especifica se usa el siguiente código

```
python manage.py test prestamos.tests.test_views.testviews.test_index_view_c
```

que corre una prueba sobre solo una pagina, en este caso index y retornara un ok si es correcta o si es un error retorna el mensaje de fallo.

Resultados de las pruebas

```
> maest > E:\> neo_library > main > python manage.py test prestamos.tests.test_views
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 10 tests in 6.065s

OK
Destroying test database for alias 'default'...
```

Se realizaron diez pruebas y todas tuvieron un resultado exitoso