

Lección 5: Actividad para portafolios nro. 5

“Implementación de autenticación y autorización en Django”

Nicolás Jofré Andrade - Desarrollo Full-stack Python

04-2024

Objetivo: Desarrollar un sistema básico de autenticación y autorización en Django, permitiendo a los usuarios iniciar sesión, cerrar sesión y gestionar permisos.

1. Configuración de autenticación:

- a. En 'settings.py', configura las URLs de redirección después de iniciar/cerrar sesión.

A screenshot of a code editor showing the Django settings.py file. The file is titled 'settings.py' with a close button. The code is as follows:

```
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'autorizacion',
41 ]
42
43 LOGIN_REDIRECT_URL = 'home'
44 LOGOUT_REDIRECT_URL = 'login'
45
```

The lines 43 and 44 are highlighted with a red rectangular box.

2. Creación del modelo de usuario:

- a. Utiliza el modelo de usuario de django para manejar la autenticación.

```
setup.py x
1 from django.contrib.auth.models import Group, Permission
2 from django.contrib.contenttypes.models import ContentType
3 from django.contrib.auth.models import User # Importa el modelo de usuario
4
5 # Obtener el usuario (por ejemplo, el primer usuario)
6 user = User.objects.first()
7
8 # Crear grupos si no existen
9 group, created = Group.objects.get_or_create(name='Admins')
10 if created:
11     # Obtener los permisos necesarios
12     content_type = ContentType.objects.get_for_model(Permission)
13     perm = Permission.objects.filter(content_type=content_type)
14     group.permissions.set(perm)
15
16 # Asignar un usuario a un grupo
17 user.groups.add(group)
18
```

- b. Ejecuta las migraciones para aplicar los cambios en la base de datos.

```
(venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod6\actividad5\autenticacion> python manage.py makemigrations
No changes detected
(venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod6\actividad5\autenticacion>
(venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod6\actividad5\autenticacion> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
(venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod6\actividad5\autenticacion>
```

3. Configuración de URLs y Vistas:

- a. Configura las URLs para las vistas de inicio de sesión y cierre de sesión en 'urls.py'

```
urls.py x
13 including another URLCONF
14 1. Import the include() function: from django.urls import include, path
15 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import include, path
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', include('autorizacion.urls'))
23 ]
24
```

b. Crea las vistas correspondientes en 'views.py'

```
views.py X
1 from django.shortcuts import render, redirect
2 from django.contrib.auth.decorators import user_passes_test
3 from django.contrib.auth import logout
4 from django.shortcuts import redirect
5
6
7 def home(request):
8     return render(request, 'home.html')
9
10 @user_passes_test(lambda u: u.is_superuser, login_url='home')
11 def admin_page(request):
12     return render(request, 'admin_page.html')
13
14 def logout_view(request):
15     logout(request)
16     return redirect('login')
```

4. Creación de grupos y permisos:

a. Define grupos y asigna permisos a través del admin de Django.

Add group

Name:

Permissions:

Available permissions ⓘ

Q Filter

- Administration | log entry | Can add log entry
- Administration | log entry | Can change log entry
- Administration | log entry | Can delete log entry
- Administration | log entry | Can view log entry
- Authentication and Authorization | permission | Can add permission
- Authentication and Authorization | permission | Can change permission
- Authentication and Authorization | permission | Can delete permission
- Authentication and Authorization | permission | Can view permission
- Authentication and Authorization | user | Can add user
- Authentication and Authorization | user | Can change user
- Authentication and Authorization | user | Can delete user

Choose all ⓘ

Chosen permissions ⓘ

Q Filter

- Authentication and Authorization | group | Can add group
- Authentication and Authorization | group | Can change group
- Authentication and Authorization | group | Can delete group
- Authentication and Authorization | group | Can view group

Remove all

Hold down "Control", or "Command" on a Mac, to select more than one.

- b. Utiliza mixins para asegurar que las vistas requieran autenticación y permisos específicos.

```
views.py X
1 from django.shortcuts import render, redirect
2 from django.contrib.auth.decorators import user_passes_test
3 from django.contrib.auth import logout
4 from django.shortcuts import redirect
5
6
7 def home(request):
8     return render(request, 'home.html')
9
10 @user_passes_test(lambda u: u.is_superuser, login_url='home')
11 def admin_page(request):
12     return render(request, 'admin_page.html')
13
14 def logout_view(request):
15     logout(request)
16     return redirect('login')
```

5. Plantillas y vistas de autenticación:

- a. Crea plantillas ('login.html', 'logout.html', 'home.html', 'admin_page.html') para las vistas y enlaces de autenticación.

