### La página de registro

A continuación, crearemos una página para que los nuevos usuarios puedan registrarse. Utilizaremos la página predeterminada de Django UserCreationForm, pero escribiremos nuestra propia función de vista y plantilla.

#### La URL de registro

El siguiente código proporciona el patrón de URL para la página de registro, que debe colocarse en *accounts/urls.py*:

**accounts/urls.py**

"""Defines URL patterns for accounts."""  
  
from django.urls import path, include  
  
from . import views  
  
app\_name = accounts  
urlpatterns = [  
 # Include default auth urls.  
 path('', include('django.contrib.auth.urls')),  
 # Registration page.  
 path('register/', views.register, name='register'),  
]

Importamos el módulo views de accounts, que necesitamos porque estamos escribiendo nuestra propia vista para la página de registro. El patrón para la página de registro coincide con la URL *http://localhost:8000/accounts/register/* y envía peticiones a la función register() que vamos a escribir.

#### La función de la vista register()

La función de vista register() tiene que mostrar un formulario de registro en blanco cuando se solicite la página de registro por primera vez, y luego procesar los formularios de registro completados cuando se envíen. Cuando un registro se realiza correctamente, la función también debe iniciar la sesión del nuevo usuario. Añade el siguiente código a *accounts/views.py*:

**accounts/views.py**

from django.shortcuts import render, redirect  
from django.contrib.auth import login  
from django.contrib.auth.forms import UserCreationForm  
  
def register(request):  
 """Register a new user."""  
 if request.method != 'POST':  
 # Display blank registration form.  
❶ form = UserCreationForm()  
 else:  
 # Process completed form.  
❷ form = UserCreationForm(data=request.POST)  
  
❸ if form.is\_valid():  
❹ new\_user = form.save()  
 # Log the user in and then redirect to home page.  
❺ login(request, new\_user)  
❻ return redirect('learning\_logs:index')  
  
 # Display a blank or invalid form.  
 context = {'form': form}  
 return render(request, 'registration/register.html', context)

Importamos las funciones render() y redirect(), y luego importamos la función login() para iniciar la sesión del usuario si su información de registro es correcta. También importamos la función por defecto UserCreationForm. En la función register(), comprobamos si estamos respondiendo a una solicitud POST. Si no es así, creamos una instancia de UserCreationForm sin datos iniciales ❶.

Si estamos respondiendo a una petición POST, creamos una instancia de UserCreationForm basada en los datos enviados ❷. Comprobamos que los datos son válidos ❸ -eneste caso, que el nombre de usuario tiene los caracteres adecuados, las contraseñas coinciden y el usuario no está intentando hacer nada malicioso en su envío.

Si los datos enviados son válidos, llamamos al método save() del formulario para guardar el nombre de usuario y el hash de la contraseña en la base de datos ❹. El método save() devuelve el objeto usuario recién creado, que asignamos a new\_user. Una vez guardada la información del usuario, iniciamos su sesión llamando a la función login() con los objetos request y new\_user ❺, lo que crea una sesión válida para el nuevo usuario. Por último, redirigimos al usuario a la página de inicio ❻, , donde un saludo personalizado en la cabecera le informa de que su registro se ha realizado correctamente.

Al final de la función, mostramos la página, que será un formulario en blanco o un formulario enviado que no es válido.

#### La plantilla de registro

Ahora crea una plantilla para la página de registro, que será similar a la página de inicio de sesión. Asegúrate de guardarla en el mismo directorio que *login.html*:

**register.html**

{% extends "learning\_logs/base.html" %}  
  
{% block content %}  
  
 <form action="{% url 'accounts:register' %}" method='post'>  
 {% csrf\_token %}  
 {{ form.as\_div }}  
  
 <button name="submit">Register</button>  
 </form>  
  
{% endblock content %}

Debería parecerse a las demás plantillas basadas en formularios que hemos estado escribiendo. Volvemos a utilizar el método as\_div para que Django muestre todos los campos del formulario adecuadamente, incluyendo cualquier mensaje de error si el formulario no se rellena correctamente.

#### Enlace a la página de registro

A continuación, añadiremos código para mostrar el enlace a la página de registro a cualquier usuario que no haya iniciado sesión:

**base.html**

--snip--  
 {% if user.is\_authenticated %}  
 Hello, {{ user.username }}.  
 {% else %}  
 <a href="{% url 'accounts:register' %}">Register</a> -  
 <a href="{% url 'accounts:login' %}">Log in</a>  
 {% endif %}  
--snip--

Ahora los usuarios que han iniciado sesión ven un saludo personalizado y un botón de cierre de sesión. Los usuarios que no han iniciado sesión ven un enlace de registro y un enlace de inicio de sesión. Prueba la página de registro creando varias cuentas de usuario con diferentes nombres de usuario.

En la siguiente sección, restringiremos algunas de las páginas para que sólo estén disponibles para los usuarios registrados, y nos aseguraremos de que cada tema pertenezca a un usuario concreto.

[anterior](c19_10.html)[Subtema 11 de 23: (Ver todo)](c19.html)[siguiente](c19_12.html)