

TEXT CLASS REVIEW

TEMAS A TRATAR EN EL CUE

- El sitio administrativo de Django.
- Registrando los modelos.
- Creando un administrador.
- Accediendo al sitio administrativo.

EL SITIO ADMINISTRATIVO DE DJANGO

INICIANDO EL SERVIDOR DE ADMINISTRACIÓN

La aplicación de administración de Django puede usar tus modelos para construir automáticamente un área dentro del sitio para crear, consultar, actualizar y borrar registros. Esto puede ahorrarte mucho tiempo de desarrollo, haciendo muy fácil probar tus modelos, y darte una idea de si tus datos son correctos. La aplicación de administración también puede ser útil para manejar datos en producción, dependiendo del estilo del sitio web.

Toda la configuración requerida para incluir la aplicación admin en tu sitio Web fue hecha automáticamente cuando [creaste el esqueleto del proyecto](#). Como resultado, lo que debes hacer para agregar tus modelos a la aplicación admin es registrarlos.

REGISTRANDO LOS MODELOS

Primero, abre `admin.py` en la aplicación catálogo (`/locallibrary/catalog/admin.py`). Actualmente se ve como esto — notar que ya importa `django.contrib.admin`:

```
1 from django.contrib import admin
2 # Register your models here.
```

Registra los modelos copiando el texto siguiente al final del archivo. Este simple código está importando los modelos, y después llama a `admin.site.register` para registrar cada uno de ellos.

```
1 from .models import Author, Genre, Book, BookInstance
2 admin.site.register(Book)
3 admin.site.register(Author)
4 admin.site.register(Genre)
5 admin.site.register(BookInstance)
```

Esta es la forma más simple de registrar un modelo, o modelos, con el sitio.

CREANDO UN ADMINISTRADOR

Para iniciar sesión en el sitio de administración, necesitamos una cuenta de usuario con estado de Personal habilitado. Para ver y crear registros, también necesitamos que este usuario tenga permisos para administrar todos nuestros objetos. Puedes crear una cuenta "administrador" que tenga acceso total al sitio, y a todos los permisos necesarios usando `manage.py`.

Usa el siguiente comando, en el mismo directorio de `manage.py`, para crear al administrador. Deberás ingresar un nombre de usuario, dirección email, y una contraseña segura.

```
1 python3 manage.py createsuperuser
```

Una vez el comando termine, un nuevo administrador será agregado a la base de datos. Ahora reinicia el servidor de desarrollo para que podamos probar el inicio de sesión:

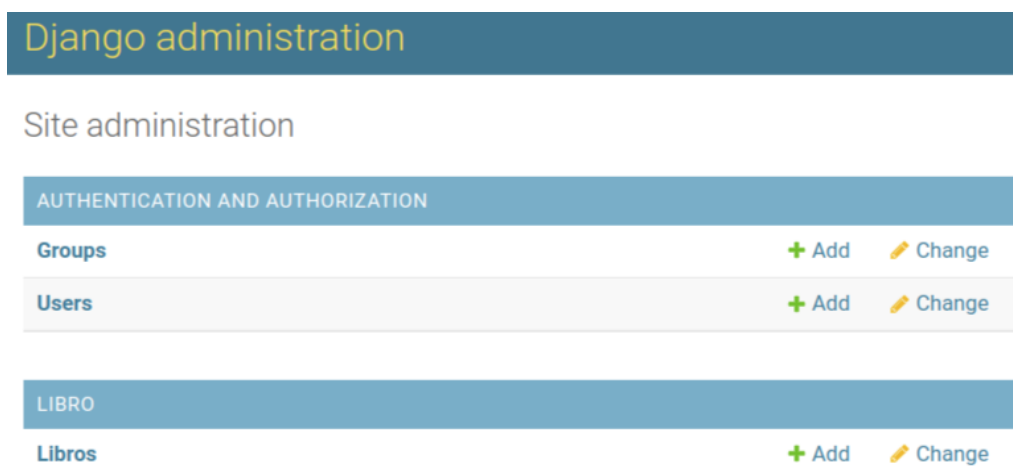
```
1 python3 manage.py runserver
```

Para iniciar sesión en el sitio, tecleamos la URL `/admin` (e.j. `http://127.0.0.1:8000/admin`), y se ingresan las credenciales de id usuario y contraseña de administrador (serás redirigido a la página login, y entonces volverás a la URL de `/admin` después de haber ingresado tus datos).

ACCEDIENDO AL SITIO ADMINISTRATIVO

El sitio muestra todos tus modelos, agrupados por aplicación instalada. Puedes hacer clic en un nombre de modelo para ir a una pantalla que lista todos los registros asociados, y además, puedes hacer clic sobre esos registros para editarlos. También puedes hacer clic directamente sobre el vínculo Añadir a continuación de cada modelo para comenzar a crear un registro de ese tipo.

Por ejemplo, si hemos construido un modelo de libros, y lo hemos registrado en el sitio de administrador de Django, podemos hacer clic sobre el vínculo Añadir a la derecha de Libros para crear uno nuevo. Ingresas valores para los campos. Se puede crear nuevos autores o géneros presionando el botón +, a continuación del campo respectivo (o seleccionar un valor existente de las listas si ya las tenías creadas). Cuando termines puedes presionar Grabar, Grabar y añadir otro, o Grabar y continuar editando para guardar el registro.



Desde esta lista puedes eliminar libros marcando la casilla de verificación junto al libro que no desees, y seleccionando la acción delete... en la lista de selección Action, y luego presionando el botón Go. Puedes también añadir nuevos libros presionando el botón ADD.

Select libro to change

Action: Delete selected libros Go 2 of 6 selected

<input type="checkbox"/>	TITULO	AUTOR	VALORACION	RATING
<input checked="" type="checkbox"/>	Django for APIs	William S. Vincent	2540	Alto
<input type="checkbox"/>	Django for Professionals	William S. Vincent	2100	Medio
<input type="checkbox"/>	Python Web Development with Django	Jeff Forcier	1350	Medio
<input type="checkbox"/>	El libro de Django	Adrian Holovaty	750	Bajo
<input type="checkbox"/>	Two Scoops of Django 3.x	Daniel Feldroy	1570	Medio
<input checked="" type="checkbox"/>	Django 3 Web Development Cookbook Fourth Edition	Aidas Bendoraitis	3250	Alto

6 libros

Puedes editar un libro haciendo clic en su nombre en la lista. La página de edición para un libro, como se muestra abajo, es casi idéntica a la página "Add". Las principales diferencias son el título de la página (Change book), y la adición de los botones Delete, HISTORY y VIEW ON SITE (este último aparece al definir el método `get_absolute_url()` al modelo), Como se refleja en: <https://docs.djangoproject.com/en/4.1/ref/models/instances/#get-absolute-url>

```
1 def get_absolute_url(self):
2     """Returns the URL to access a particular instance of the model."""
3     return reverse('model-detail-view', args=[str(self.id)])
```

Change libro

HISTORY

Django for APIs

Título:

Django for APIs

Autor:

William S. Vincent

Valoracion:

2540

Valor entre 0 y 10000

Fecha creacion:

Aug. 16, 2022, 2:32 p.m.

Fecha modificacion:

Aug. 16, 2022, 2:32 p.m.

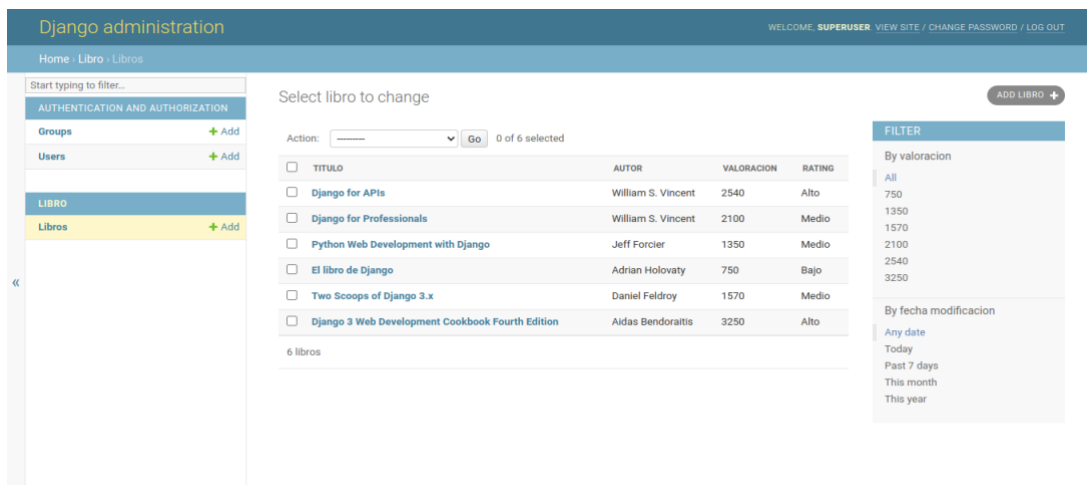
Delete

Save and add another

Save and continue editing

SAVE

Ahora, regresa a la página Home (usando el enlace Home de la barra superior), y observa las listas de **Título**, **Autor**, **Valoración** y **Rating** que se han creado previamente. A partir de este administrador de Django se pueden hacer gestiones de manipulación de modelos como CRUD: crear, recuperar, actualizar y eliminar respectivamente.



PROBANDO USUARIOS EN EL MODELO AUTH

Django proporciona un sistema de autenticación y autorización ("permisos"), construido sobre el framework de sesión, que le permite verificar credenciales de usuario y definir qué acciones puede realizar cada uno. El framework incluye modelos para **Users** y **Groups** (una forma genérica de aplicar permisos a más de un usuario a la vez), permisos/indicadores (permissions/flags) que designan si un usuario puede realizar una tarea, formularios y vistas para iniciar sesión en los usuarios, y **view tools** para restringir el contenido.

El sistema de autenticación es muy flexible, y puedes crear tus URLs, formularios, vistas y plantillas desde el inicio, simplemente llamando a la API provista para loguear al usuario. La configuración se establece en las secciones **INSTALLED_APPS** y **MIDDLEWARE** del archivo del proyecto (**locallibrary/locallibrary/settings.py**), tal como se muestra abajo:

```
1 INSTALLED_APPS = [  
2     ...  
3     'django.contrib.auth', #Core authentication framework and its default  
4     models.  
5     'django.contrib.contenttypes', #Django content type system (allows  
6     permissions to be associated with models).  
7     ....  
8  
9     MIDDLEWARE = [  
10        ...  
11        'django.contrib.sessions.middleware.SessionMiddleware', #Manages  
12        sessions across requests  
13        ...  
14        'django.contrib.auth.middleware.AuthenticationMiddleware',  
15        #Associates users with requests using sessions.  
16        ....
```

MANEJANDO ERRORES EN EL MODELO AUTH

Un gran error que nos podemos enfrentar en el modelo auth es el no limitar el acceso a cierto contenido. La forma básica de limitar el acceso a las páginas es verificar `request.user.is_authenticated` y redirigir a la página de login:

```
1 from django.conf import settings  
2 from django.shortcuts import redirect  
3  
4 def mi_vista(request):  
5     if not request.user.is_authenticated:  
6         return redirect('%s?next=%s' % (settings.LOGIN_URL, request.path))
```

Otra forma de limitar el acceso es mostrando un mensaje de error:

```
1 from django.shortcuts import render  
2  
3 def mi_vista(request):  
4     if not request.user.is_authenticated:  
5         return render(request, 'myapp/login_error.html')
```

ADMINISTRANDO USUARIOS Y GRUPOS

El superusuario ya está autenticado y tiene todos los permisos, así que se necesita crear un usuario de prueba que represente a un usuario normal del sitio.

```
1 from django.contrib.auth.models import User
2
3 # Create user and save to the database
4 user = User.objects.create_user('myusername', 'myemail@crazymail.com',
5 'mypassword')
6
7 # Update fields and then save again
8 user.first_name = 'John'
9 user.last_name = 'Citizen'
10 user.save()
```

A continuación, primero crearemos un grupo y luego un usuario. Aunque no tengamos ningún permiso aún para otorgar a los miembros de nuestra biblioteca, si lo necesitamos más adelante, será mucho más fácil otorgarlo una vez al grupo que individualmente a cada miembro.

Inicia el servidor de desarrollo y navega hasta el sitio de administración en tu navegador web local (<http://127.0.0.1:8000/admin/>). Ingresa al sitio usando las credenciales de la cuenta de tu superusuario. El nivel superior del sitio de administración "Admin site" muestra todos tus modelos, los cuales están ordenados por la aplicación por defecto de Django "django application". Desde la sección de Autenticación y Autorización puedes dar clic en los enlaces de Usuarios **"Users"** y Grupos **"Groups"** para ver todos sus registros existentes.

Primero vamos a crear un nuevo grupo para los miembros de nuestra biblioteca. Da clic en el botón Add "Añadir" (Enseguida de Group) para crear un nuevo grupo; ingresa el Nombre "Name" para el grupo de "Library Members", **admin site - add Group**. No necesitamos de ningún permiso para el grupo, entonces solo presiona Save (Serás redirigido a una lista de los grupos disponibles).

Ahora vamos a crear un usuario: navega de vuelta a la página de inicio **"home"** del sitio de administración "Admin site". Da clic en el botón **Add** "Añadir" que queda enseguida de **Users** "Usuarios", para abrir el cuadro de diálogo de Usuario **Add** "Añadir usuario". Ingresa un Nombre de Usuario **"Username"**, Contraseña

"Password", y Confirmación de Contraseña **"Password confirmation"** apropiado para tu usuario de prueba. Presiona **Save** "Guardar" para crear el usuario.

MANEJO DE PERMISOS POR USUARIO EN PÁGINA

El sitio de administrador creará el nuevo usuario, e inmediatamente te llevará a la pantalla de **Change user** "Cambios del usuario", en la que puedes cambiar tu nombre de usuario **"Username"**, y agregar información para los campos opcionales del modelo de Usuario "User". Estos campos incluyen el primer nombre **"first name"**, el apellido **"last name"**, la dirección de correo electrónico **"email adress"**, los estados de los usuarios y sus permisos "users status and permissions" (solo el indicador **Active** "Activo" debería ser activado). Más abajo puedes especificar los grupos y permisos del usuario, y revisar datos importantes relacionados al usuario (ejemplo: la fecha en que se agregó y la fecha del último inicio de sesión).

MANEJO DE GRUPOS EN PÁGINA ADMINISTRATIVA DE DJANGO

En la sección **Groups** "Grupos", selecciona el grupo **Library Member** de la lista de grupos disponibles, y entonces presiona la flecha apuntando a la derecha entre las dos cajas para moverlo dentro de la caja de **Chosen groups** "Grupos seleccionados". Aquí no necesitamos hacer nada adicional, por lo que de nuevo solo selecciona **SAVE** "Guardar", para ir a la lista de usuarios.