

Contenidos a Trabajar

1. El sitio de administración de Django

- Resumen
- Asociando instancias AdminSite al URLconf
- Objetos ModelAdmin

El sitio de administración de Django

Una de las partes más poderosas de Django es la interfaz de administración automática. Lee los metadatos de sus modelos para proporcionar una interfaz rápida y centrada en el modelo donde los usuarios de confianza pueden administrar el contenido de su sitio. El uso recomendado por el administrador se limita a la herramienta de administración interna de una organización. No está diseñado para construir todo el front-end.

El administrador tiene muchos puntos para la personalización, pero tenga cuidado de intentar usar esos puntos exclusivamente. Si necesita proporcionar una interfaz más centrada en el proceso que abstraiga los detalles de implementación de las tablas y los campos de la base de datos, probablemente sea el momento de describir sus propias vistas.

En este documento veremos los puntos más importantes de la interfaz de administración de Django.

Resumen

El administrador está habilitado en la plantilla de proyecto predeterminada utilizada por **startproject**. Si no está utilizando la plantilla de proyecto predeterminada, estos son los requisitos:

1. Agregue '**django.contrib.admin**' y sus dependencias - **django.contrib.auth**, **django.contrib.contenttypes**, **django.contrib.messages** y **django.contrib.sessions**- a su configuración **INSTALLED_APPS**.
2. Configura un backend **DjangoTemplates** en la configuración **TEMPLATES** con **django.template.context_processors.request**, **django.contrib.auth.context_processors.auth** y **django.contrib.messages.context_processors.messages** en la opción '**context_processors**' de **OPTIONS**.
3. Si ha personalizado la configuración **MIDDLEWARE**, **django.contrib.auth.middleware.AuthenticationMiddleware** y **django.contrib.messages.middleware.MessageMiddleware** deben incluirse.
4. Asocie las URL del administrador en su URLconf.

Una vez que haya realizado estos pasos, podrá usar el sitio de administración visitando la URL a la que loconectó (**/admin/**, de forma predeterminada).

Si necesita crear un usuario para iniciar sesión, use el comando **createsuperuser**. De forma predeterminada, iniciar sesión en el administrador requiere que el usuario tenga el atributo **is_staff** establecido en **True**.

Finalmente, determine cuál de los modelos de su aplicación debe ser editable en la interfaz de administración. Para cada uno de esos modelos, regístrelos con el administrador como se describen **ModelAdmin** mas adelante.

Asociando instancias AdminSite al URLconf

El paso para configurar el administrador de Django es conectar su instancia **AdminSite** a su URLconf. Haga esto apuntando una URL determinada al método **AdminSite.urls**. No es necesario usar **include()**.

En este ejemplo, registramos la instancia predeterminada **django.contrib.admin.site** del **AdminSite** en la URL **/admin/**

```
# urls.py
from django.contrib import admin
from django.urls import path

urlpatterns = [
    path('admin/', admin.site.urls),
```

Objetos ModelAdmin

Clase ModelAdmin

La clase **ModelAdmin** es la representación de un modelo en la interfaz de administración. Por lo general, estos se almacenan en un archivo nombrado **admin.py** en su aplicación. Veamos un ejemplo de **ModelAdmin**:

```
from django.contrib import admin
from myapp.models import Author

class AuthorAdmin(admin.ModelAdmin):
    pass
```



```
admin.site.register (Author, AuthorAdmin)
```

En el ejemplo anterior, la clase **ModelAdmin** no define ningún valor personalizado (todavía). Como resultado, se proporcionará la interfaz de administración predeterminada. Si está satisfecho con la interfaz de administración predeterminada, no necesita definir ningún objeto **ModelAdmin**; puede registrar la clase modelo sin proporcionar una descripción **ModelAdmin**. El ejemplo anterior podría simplificarse a:

```
from django.contrib import admin
from myapp.models import Author

admin.site.register (Author)
```

El **ModelAdmin** es muy flexible. Tiene varias opciones para lidiar con la personalización de la interfaz. Todas las opciones se definen en la subclase **ModelAdmin**.

Para revisar más opciones de configuración para los **ModelAdmin** o formas de registrar modelos, como pueden ser los decoradores, por favor revise la documentación oficial actualizada:

<https://docs.djangoproject.com/es/3.2/ref/contrib/admin/>

En la documentación, verá configuraciones simples como `list_display`, donde podrá controlar que campos desea mostrar en la página del administrador de un modelo. Si no se establece `list_display`, el sitio de administración mostrará una sola columna que muestra la representación `__str__()` de cada objeto.