

Proyecto Django: Base de Datos de Países y Ciudades

Este documento describe los archivos añadidos y modificados para crear una base de datos de países y ciudades en Django, cómo cargar los datos desde python manage.py shell y cómo solucionar el problema de error 404 al acceder a la raíz del proyecto.

Archivos Añadidos y Modificados

1. Creación de la Aplicación locations

Ejecutar el siguiente comando para crear la aplicación:

```
python manage.py startapp locations
```

2. Agregar locations a INSTALLED_APPS en settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'locations', # Se agrega la nueva aplicación  
]
```

3. Definición de Modelos en locations/models.py

```
from django.db import models  
  
class Pais(models.Model):  
    nombre = models.CharField(max_length=100)  
  
    def __str__(self):  
        return self.nombre  
  
class Ciudad(models.Model):  
    nombre = models.CharField(max_length=100)  
    pais = models.ForeignKey(Pais, on_delete=models.CASCADE)  
  
    def __str__(self):  
        return f"{self.nombre}, {self.pais.nombre}"
```

4. Crear y Aplicar Migraciones

```
python manage.py makemigrations locations python manage.py migrate
```

5. Registrar Modelos en locations/admin.py

```
from django.contrib import admin
from .models import Pais, Ciudad

admin.site.register(Pais)
admin.site.register(Ciudad)
```

6. Crear Vista en locations/views.py

```
from django.shortcuts import render
from .models import Pais

def lista_paises(request):
    paises = Pais.objects.all()
    return render(request, "locations/paises.html", {"paises": paises})
```

7. Configuración de URLs en locations/urls.py

```
from django.urls import path
from .views import lista_paises

urlpatterns = [
    path('', lista_paises, name='lista_paises'),
]
```

8. Configuración de URLs en GeoApp/urls.py (Solución al Error 404)

```
from django.contrib import admin
from django.urls import path, include
from django.shortcuts import redirect

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', lambda request: redirect('lista_paises')), # Redirigir la raíz a /paises/
    path('paises/', include('locations.urls')),
]
```

9. Crear Plantilla locations/templates/locations/paises.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Lista de Países</title>
</head>
<body>
  <h1>Lista de Países</h1>
  <ul>
    {% for pais in paises %}
      <li>{{ pais.nombre }}</li>
    {% endfor %}
  </ul>
</body>
</html>
```

Cargar Datos en la Base de Datos

Para agregar datos a la base de datos, se puede usar el shell de Django:

Abrir el shell de Django:

```
python manage.py shell
```

Ejecutar los siguientes comandos en el shell:

```
from locations.models import Pais, Ciudad
```

```
# Crear países
colombia = Pais.objects.create(nombre="Colombia")
mexico = Pais.objects.create(nombre="México")

# Crear ciudades
Ciudad.objects.create(nombre="Bogotá", pais=colombia)
Ciudad.objects.create(nombre="Medellín", pais=colombia)
Ciudad.objects.create(nombre="Ciudad de México", pais=mexico)
Ciudad.objects.create(nombre="Guadalajara", pais=mexico)
```

Salir del shell con el comando:

```
exit()
```

Problemática y Solución Aplicada

Problema: Error 404 al acceder a la raíz del sitio (http://127.0.0.1:8000/)

Mensaje de error:

Page not found (404) Request URL: http://127.0.0.1:8000/ Using the URLconf defined in GeoApp.urls, Django tried these URL patterns, in this order:

admin/ paises/ [name='lista_paises'] The empty path didn't match any of these.

Solución: Redirigir la raíz a /paises/

Se modificó el archivo GeoApp/urls.py agregando una redirección:

```
from django.contrib import admin
from django.urls import path, include
from django.shortcuts import redirect

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', lambda request: redirect('lista_paises')), # Redirigir la raíz a
/paises/
    path('paises/', include('locations.urls')),
]
```

Con esta solución, ahora cuando un usuario accede a http://127.0.0.1:8000/, automáticamente es redirigido a http://127.0.0.1:8000/paises/, evitando el error 404.