66_Lavanya Kini

Django REST Framework - Library Management System

1. Install Dependencies

```
pip install django djangorestframework
```

2. Create Django Project and App

```
django-admin startproject library_project

cd library_project

django-admin startapp library_app
```

3. Configure Django Settings (library_project/settings.py)

```
INSTALLED_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'rest_framework',

    'library_app',

]
```

4. Define Models (library_app/models.py)

```
from django.db import models


class Author(models.Model):

    id = models.AutoField(primary_key=True)

    name = models.CharField(max_length=100)

    birth_date = models.DateField(null=True, blank=True)

    def __str__(self): return self.name


class Book(models.Model):

    id = models.AutoField(primary_key=True)
    title = models.CharField(max_length=200)
    isbn = models.CharField(max_length=13, unique=True)
    author = models.ForeignKey(Author, on_delete=models.CASCADE, related_name='books')
    def __str__(self): return self.title
```

5. Run Migrations

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

6. Create Serializers (library_app/serializers.py)

```
from rest_framework import serializers
from .models import Author, Book

class AuthorSerializer(serializers.ModelSerializer):
    class Meta:
        model = Author
        fields = '__all__'

class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        fields = '__all__'
```

7. Create Views (library_app/views.py)

```
from rest_framework import generics
from .models import Book, Author
from .serializers import BookSerializer, AuthorSerializer

class AuthorListCreateView(generics.ListCreateAPIView):
    queryset = Author.objects.all()
    serializer_class = AuthorSerializer

class AuthorRetrieveUpdateDestroyView(generics.RetrieveUpdateDestroyAPIView):
    queryset = Author.objects.all()
    serializer_class = AuthorSerializer


class BookListCreateView(generics.ListCreateAPIView):
    queryset = Book.objects.all()
    serializer_class = BookSerializer


class BookRetrieveUpdateDestroyView(generics.RetrieveUpdateDestroyAPIView):
    queryset = Book.objects.all()
    serializer_class = BookSerializer
```

8. Define URLs (library_app/urls.py)

```
from django.urls import path
from .views import AuthorListCreateView, AuthorRetrieveUpdateDestroyView,
BookListCreateView, BookRetrieveUpdateDestroyView


urlpatterns = [
    path('authors/', AuthorListCreateView.as_view(), name='author-list'),
    path('authors/<int:pk>/', AuthorRetrieveUpdateDestroyView.as_view(),
name='author-detail'),
    path('books/', BookListCreateView.as_view(), name='book-list'),
    path('books/<int:pk>/', BookRetrieveUpdateDestroyView.as_view(),
name='book-detail'),
]
```

9. Register URLs in Project (library_project/urls.py)

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


urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('library_app.urls')),
]
```

10. Run Server

```
python manage.py runserver
```

11. Test API

```
curl -X POST http://127.0.0.1:8000/api/authors/ -H "Content-Type: application/json" -d
'{"name": "J.K. Rowling", "birth_date": "1965-07-31"}'

curl -X POST http://127.0.0.1:8000/api/books/ -H "Content-Type: application/json" -d
'{"title": "Harry Potter", "isbn": "1234567890123", "author": 1}'

curl -X GET http://127.0.0.1:8000/api/books/
```