# Django REST Framework (DRF) - Book Management API Project

## 1. Project Setup

Create a Django project and app:

```
$ django-admin startproject BookAPI
$ cd BookAPI
$ python manage.py startapp books
```

Add 'rest_framework' and 'books' to INSTALLED_APPS in settings.py.

Run:

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

## 2. Model (books/models.py)

```
from django.db import models


class Book(models.Model):
    title = models.CharField(max_length=255)
    author = models.CharField(max_length=255)
    published_date = models.DateField()
    isbn = models.CharField(max_length=13)


    def __str__(self):
        return self.title
```

## 3. Serializer (books/serializers.py)

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


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

## 4. Views (books/views.py)

```
from rest_framework import viewsets
from .models import Book
from .serializers import BookSerializer

class BookViewSet(viewsets.ModelViewSet):
    queryset = Book.objects.all()
    serializer_class = BookSerializer
```

## 5. URLs (books/urls.py)

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import BookViewSet

router = DefaultRouter()
router.register(r'books', BookViewSet)

urlpatterns = [
    path('', include(router.urls)),
]
```

## 6. Project URLs (BookAPI/urls.py)

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

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

## 7. Running the Server

```
$ python manage.py runserver
```

API Endpoints:

- POST /api/books/       : Create a new book
- GET /api/books/        : List all books
- GET /api/books/{id}/   : Retrieve specific book
- PUT /api/books/{id}/   : Update a specific book
- DELETE /api/books/{id}/: Delete a book

## 8. Screenshots of API Endpoints (Simulated)

**Insert a Book Record (POST)**

```
POST /api/books/
Request Body:
{
  "title": "Django Basics",
  "author": "John Doe",
  "published_date": "2024-01-01",
  "isbn": "1234567890123"
}
Response: 201 Created
```

**View All Books (GET)**

```
GET /api/books/
Response: 200 OK
[
  {
    "id": 1,
    "title": "Django Basics",
    ...
  }
]
```

**View Book by ID (GET)**

```
GET /api/books/1/
Response: 200 OK
{
  "id": 1,
  "title": "Django Basics",
  ...
}
```

**Update Book Record (PUT)**

```
PUT /api/books/1/
Request Body:
{
  "title": "Django Advanced",
  ...
}
Response: 200 OK
```

**Delete Book Record (DELETE)**

```
DELETE /api/books/1/
Response: 204 No Content
```