



สร้าง API ง่ายๆ ด้วย Django REST Framework

คู่มือฉบับรวมรัด จบใน 15 สไลด์ จากแนวคิดสู่การใช้งานจริง

เหมาะสำหรับนักพัฒนาที่ต้องการเข้าใจพื้นฐานการสร้าง REST API

Django REST Framework (DRF) คืออะไร?

DRF คือส่วนขยาย (Extension) ของ Django ที่ช่วยแปลง Database Models ให้กลายเป็น RESTful API เพื่อให้ Web Apps หรือ Mobile Apps อื่นๆ สามารถเรียกใช้งานข้อมูลได้ง่ายดาย



3 องค์ประกอบหลักที่คุณต้องรู้



1. Serialization

ตัวแปลงภาษา – แปลงข้อมูล
ชั้บช้อน (Django Models)
ให้เป็น JSON หรือ XML



2. Viewsets

ตัวจัดการคำขอ – กำหนด
Logic ในการรับและตอบ
ตอบกลับ API Requests



3. URL Routing

แผนที่การเดินทาง – เชื่อม
โยง URL ไปยัง API Views
เพื่อให้ Client เข้าถึงได้

ໂຈຍຂອງເຮົາ: ຮະບບຈັດກາຣໜັງສື່ວ

ເຮົາຈະສ້າງ CRUD API (Create, Read, Update, Delete) ສໍາຮັບຈັດກາຣໜັງສື່ວ
ໂດຍມີໂຄຮງສ້າງຂໍ້ອມຸລດັ່ງນີ້

Model: Book

-  title: ຜົນເປົ້າ (String)
-  author: ຜູ້ແຕ່ງ (String)
-  publish_date: ວັນທີເຕີພິມ (Date)

ขั้นตอนที่ 1: ติดตั้งและตั้งค่า

ติดตั้งライบรารีผ่าน Command Line:

```
pip install djangorestframework
```

เพิ่มลงใน settings.py:

ต้องเพิ่มทั้ง 'rest_framework' และ 'app'
ของเราลงใน INSTALLED_APPS

```
1  
2 INSTALLED_APPS = [  
3     ...  
4     'rest_framework',  
5     'app',  
6 ]  
7
```

ขั้นตอนที่ 2: สร้าง Book Model

File: app/models.py

```
from django.db import models

class Book(models.Model):
    title = models.CharField(max_length=200)
    author = models.CharField(max_length=100)
    publish_date = models.DateField()

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

≡ เก็บข้อความสั้น

📅 เก็บวันที่

⌚ แสดงผลชื่อหนังสือให้อ่านง่าย

ขั้นตอนที่ 3: สร้าง Serializer

File: app/serializers.py

- กำหนดที่แปลง Model Instance ให้เป็น JSON และตรวจสอบความถูกต้องของข้อมูล

Key Takeaway: ModelSerializer ช่วยสร้าง Serializer ให้อัตโนมัติตามฟิลด์ที่เรากำหนด

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

class BookSerializer(serializers.ModelSerializer):

    class Meta:
        model = Book
        fields = ['id', 'title', 'author', 'publish_date']
```

ขั้นตอนที่ 4: กำหนด ViewSet

File: app/views.py

- ⊕ รวมคำสั่ง CRUD (สร้าง, อ่าน, แก้ไข, ลบ) ไว้ในที่เดียว

Insight: ModelViewSet
คือทางลัดที่จัดการ Logic
พื้นฐานให้เราทึ่งหมดกันที

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

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

Magic happens here! 

ขั้นตอนที่ 5: สร้าง URL ด้วย Router

File: app/urls.py

ไม่ต้องกำหนด URL เองทีละตัว Router จะสร้าง Standard Endpoints ให้เอง

```
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)),
]
```

ขั้นตอนที่ 6: เชื่อมต่อกับ Project หลัก

File: project/urls.py

นำ URLs ของแอปไปต่อท้าย
path 'api/' ในไฟล์หลัก

Result: URL ก็งหมดจะขึ้นต้น
ด้วย /api/

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

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

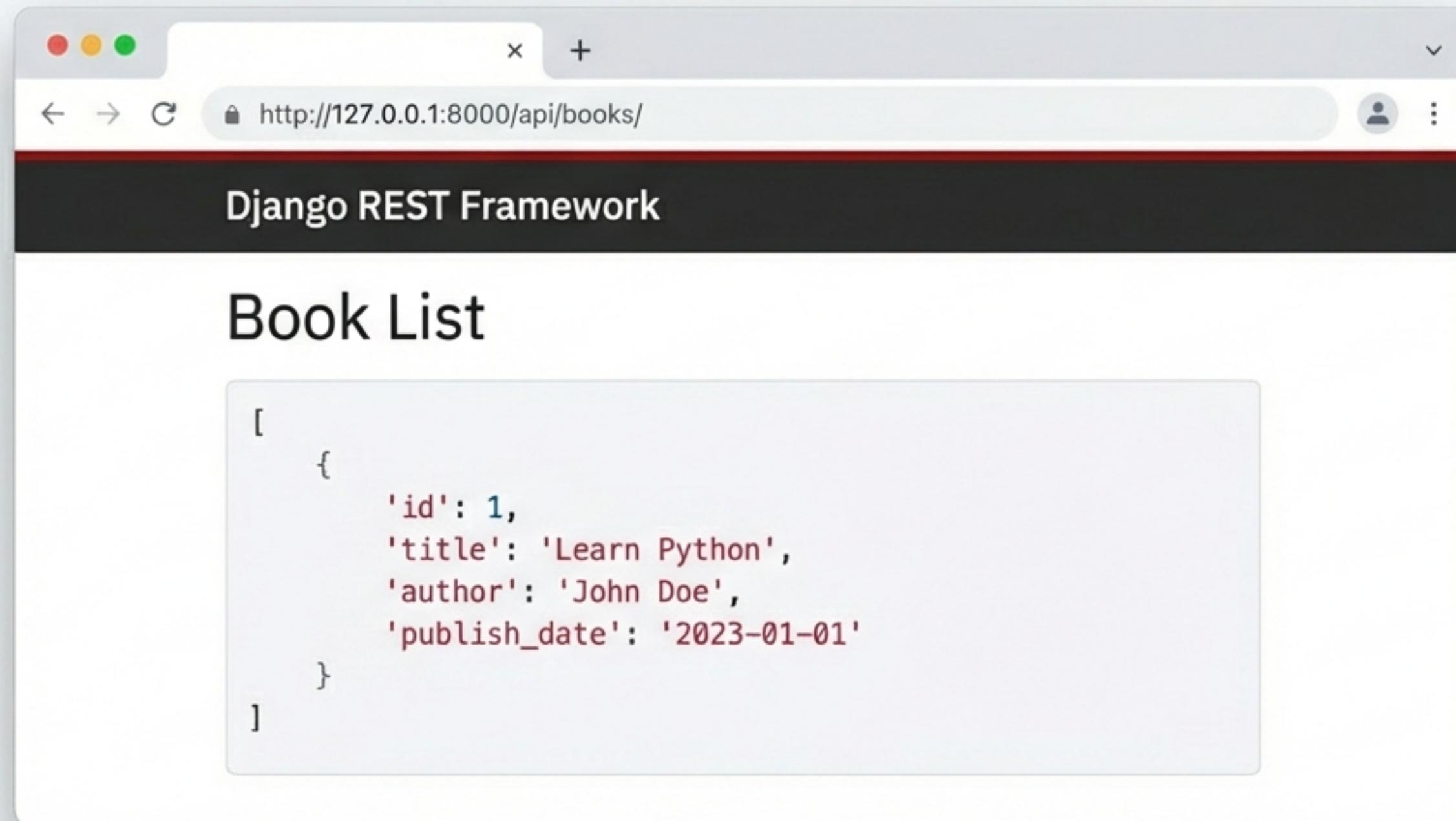
ขั้นตอนที่ 7: อัปเดตฐานข้อมูลและรันเซิร์ฟเวอร์

```
$ python manage.py makemigrations
Migrations for 'app': Created model Book
$ python manage.py migrate
Applying app.0001_initial... OK
$ python manage.py runserver
Starting development server at http://127.0.0.1:8000/
```

1. makemigrations: เตรียมไฟล์สำหรับสร้างตาราง
2. migrate: สร้างตารางจริงในฐานข้อมูล
3. runserver: เริ่มต้นระบบ

ผลลัพธ์: The Browsable API

เข้าถึงได้ผ่าน Browser ที่ <http://127.0.0.1:8000/api/books/>



The screenshot shows a web browser window with the URL <http://127.0.0.1:8000/api/books/>. The page title is "Django REST Framework" and the main content is "Book List". Below the title, there is a JSON response for a single book:

```
[  
  {  
    "id": 1,  
    "title": "Learn Python",  
    "author": "John Doe",  
    "publish_date": "2023-01-01"  
  }]
```



DRF มีหน้าเว็บสำเร็จรูปให้เราทดสอบ API ได้กันที่โดยไม่ต้องเขียน Frontend

การจัดการข้อมูล (Data Interaction)

- หน้าเว็บ Browsable API อนุญาตให้เราส่งข้อมูลได้โดยตรง:
 - POST: กรอกฟอร์มด้านล่างเพื่อ 'สร้างหนังสือใหม่'
 - GET (id): เข้าไปที่ /api/books/1/ เพื่อดูเล่มเดียว
 - PUT/DELETE: ปุ่มกดเพื่อแก้ไข หรือลบข้อมูล

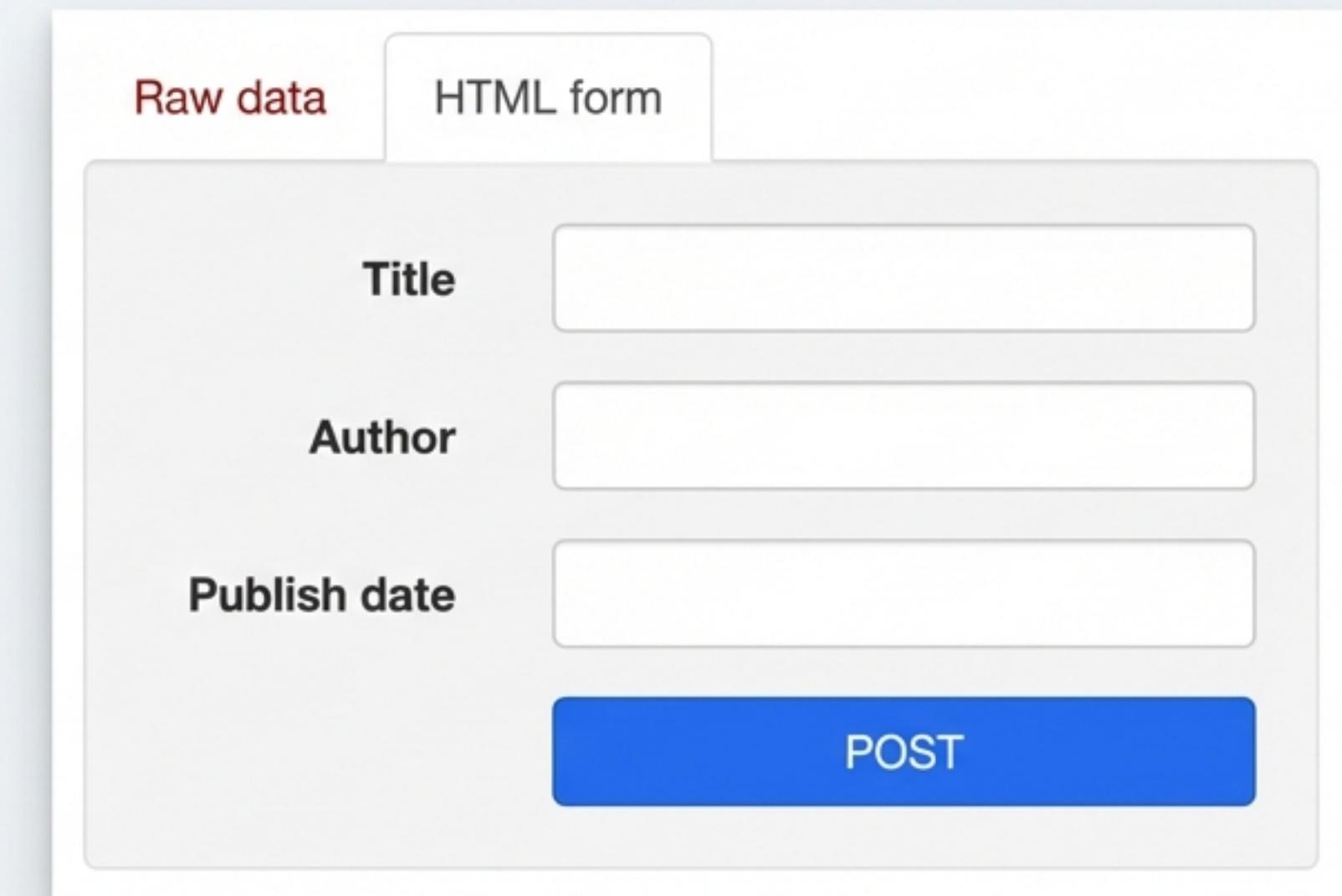
Raw data HTML form

Title

Author

Publish date

POST



สรุป Endpoints ที่พร้อมใช้งาน

HTTP Method	URL	Description
GET	/api/books/	ดูรายการหนังสือทั้งหมด
POST	/api/books/	สร้างหนังสือใหม่
GET	/api/books/{id}/	ดูรายละเอียดหนังสือ 1 เล่ม
PUT	/api/books/{id}/	แก้ไขข้อมูลหนังสือ
DELETE	/api/books/{id}/	ลบหนังสือ

บทสรุป

ยินดีด้วย! คุณสร้าง REST API ที่สมบูรณ์แบบเสร็จเรียบร้อยแล้ว

ตอนนี้คุณสามารถเชื่อมต่อ API นี้เข้ากับ React, Vue, Mobile App หรือระบบอื่นๆ ได้ทันที

เนื้อหาอ้างอิงจากบทความ 'Building a Simple API with Django REST Framework' โดย GeeksforGeeks