

Owais Dhukka 13

Assignment 4 - Python Integration Primer

```
In [ ]: ### models.py  
from django.db import models  
  
class Author(models.Model):  
    name = models.CharField(max_length=100)  
  
    def __str__(self):  
        return self.name  
  
class Book(models.Model):  
    title = models.CharField(max_length=200)  
    isbn = models.CharField(max_length=13, unique=True)  
    author = models.ForeignKey(Author, on_delete=models.CASCADE)  
  
    def __str__(self):  
        return self.title
```

```
In [ ]: ### 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__'
```

```
In [ ]: ### urls.py  
from django.urls import path  
from .views import AuthorListCreateView, AuthorDetailView, BookListCreateView  
  
urlpatterns = [  
    path('authors/', AuthorListCreateView.as_view(), name='author-list-create'),  
    path('authors/<int:pk>/', AuthorDetailView.as_view(), name='author-detail'),  
    path('books/', BookListCreateView.as_view(), name='book-list-create'),  
    path('books/<int:pk>/', BookDetailView.as_view(), name='book-detail'),  
]
```

```
In [ ]: ### views.py  
from django.shortcuts import render
```

```

from rest_framework import generics
from .models import Author, Book
from .serializers import AuthorSerializer, BookSerializer
from django.http import JsonResponse

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

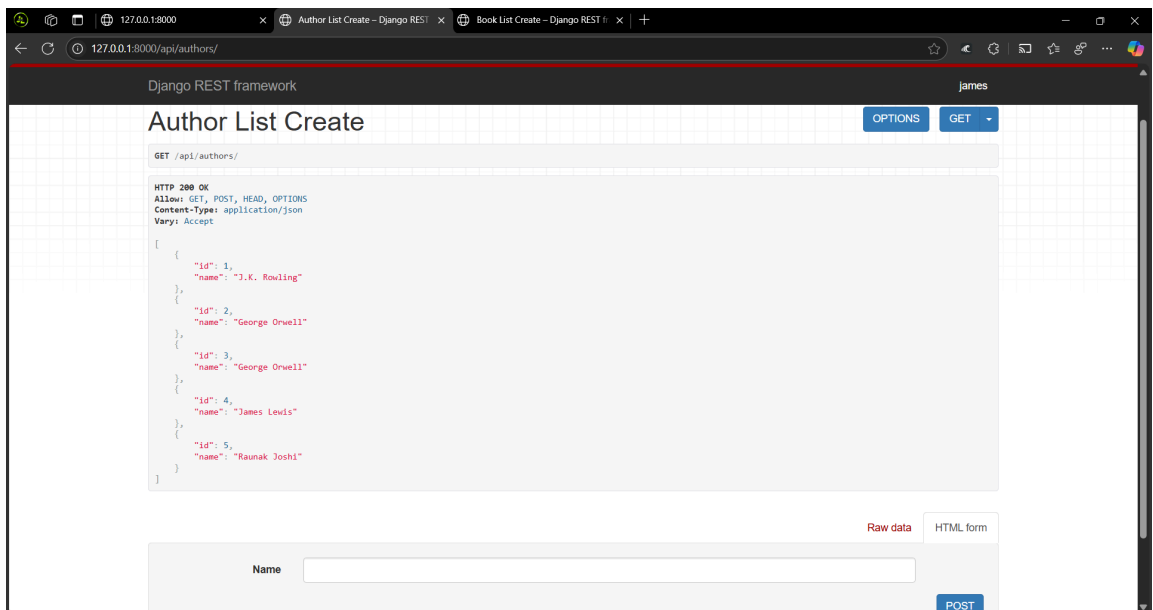
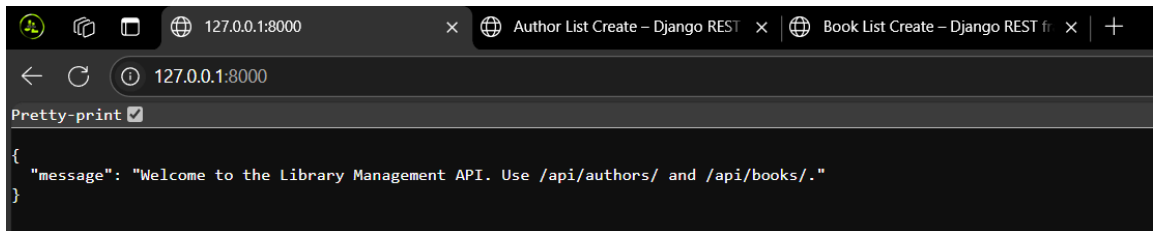
class AuthorDetailView(generics.RetrieveUpdateDestroyAPIView):
    queryset = Author.objects.all()
    serializer_class = AuthorSerializer

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

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

def home_view(request):
    return JsonResponse({"message": "Welcome to the Library Management API."})

```



Django REST framework

Book List Create

OPTIONS GET

GET /api/books/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
[
  {
    "id": 1,
    "title": "Harry Potter",
    "isbn": "1234567890123",
    "author": 1
  },
  {
    "id": 2,
    "title": "Harry Potter and the Sorcerer's Stone",
    "isbn": "1234567890122",
    "author": 1
  }
]
```

Raw data HTML form

Title

Isbn

Author

POST

This notebook was converted with convert.ploomber.io