Open xampp and run apache and mysql server

Go to phpmyadmin

Create new DB named django\_rest\_book\_api

Now Create Django Project

django-admin startproject book\_api

in vs code

python manage.py startapp api

Connect App

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'api.apps.ApiConfig',

    'rest\_framework',

]

Install Django rest Framwork

pip install djangorestframework

add database to settings.py

# Database

# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': 'django\_rest\_book\_api',

        'USER': 'root',

        'PASSWORD': '',

        'HOST': '127.0.0.1',

        'PORT': '3306',

    }

}

Models.py

from django.db import models

# Create your models here.

class Book(models.Model):

    title = models.CharField(max\_length=200, null=False, blank=False)

    genre = models.CharField(max\_length=100, null=False, blank=False)

    price = models.DecimalField(max\_digits=4, decimal\_places=2)

    summary = models.TextField()

    stars=models.IntegerField()

    def \_\_str\_\_(self):

        return self.name

url.py(root)

"""book\_api URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

    https://docs.djangoproject.com/en/4.0/topics/http/urls/

Examples:

Function views

    1. Add an import:  from my\_app import views

    2. Add a URL to urlpatterns:  path('', views.home, name='home')

Class-based views

    1. Add an import:  from other\_app.views import Home

    2. Add a URL to urlpatterns:  path('', Home.as\_view(), name='home')

Including another URLconf

    1. Import the include() function: from django.urls import include, path

    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('api/', include('api.urls')),

]

url.py

from django.urls import path

from api import views

urlpatterns = [

    path('', views.apiOverview, name='apiOverview'),

]

Views.py

from django.shortcuts import render

from rest\_framework.response import Response

from rest\_framework.decorators import api\_view

# Create your views here.

@api\_view(['GET'])

def apiOverview(request):

    api\_urls = {

        'List': '/book-list/',

        'Detail-view': '/book-detail/<int:id>',

        'Create': '/book-create/',

        'Update-view': '/book-update/<int:id>',

        'Delete-view': '/book-delete/<int:id>',

    }

    return Response(api\_urls)

Install mysqlclient

pip install mysqlclient

admin.py

import imp

from django.contrib import admin

from .models import Book

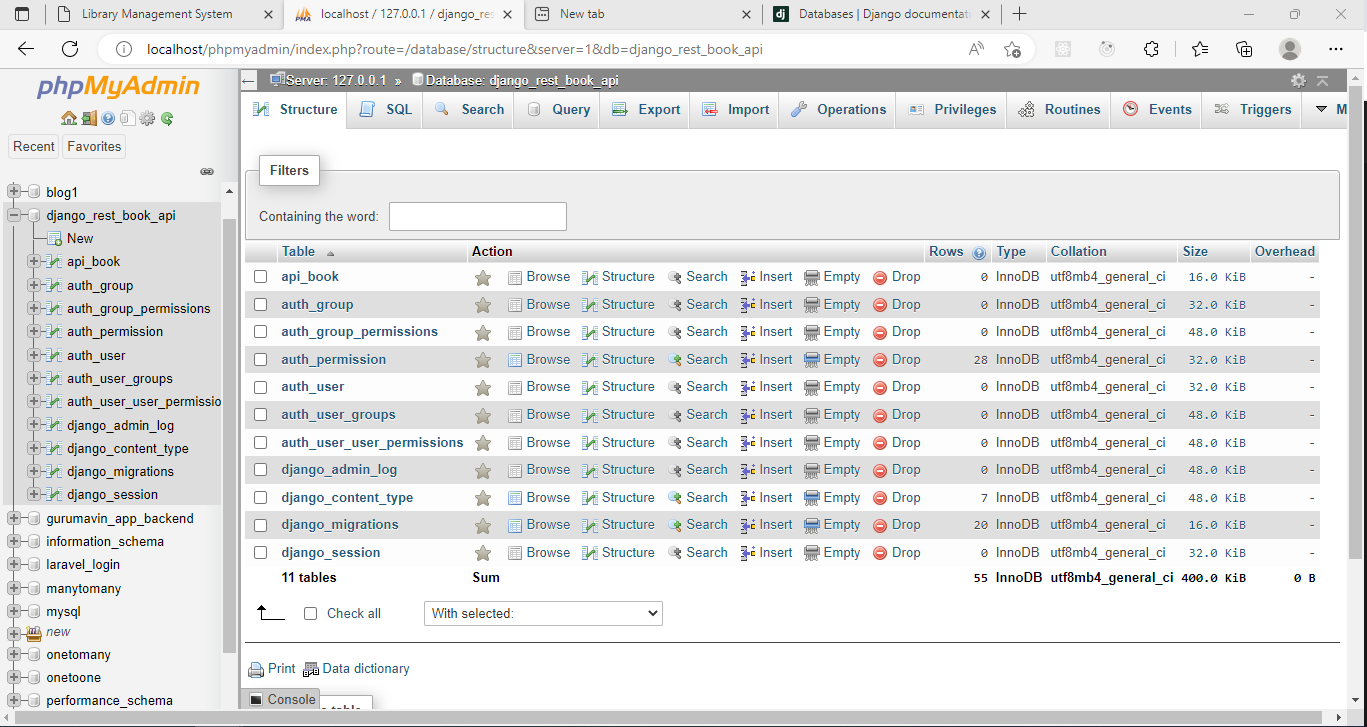
# Register your models here.

admin.site.register(Book)

Migrations

python manage.py makemigrations

python manage.py migrate

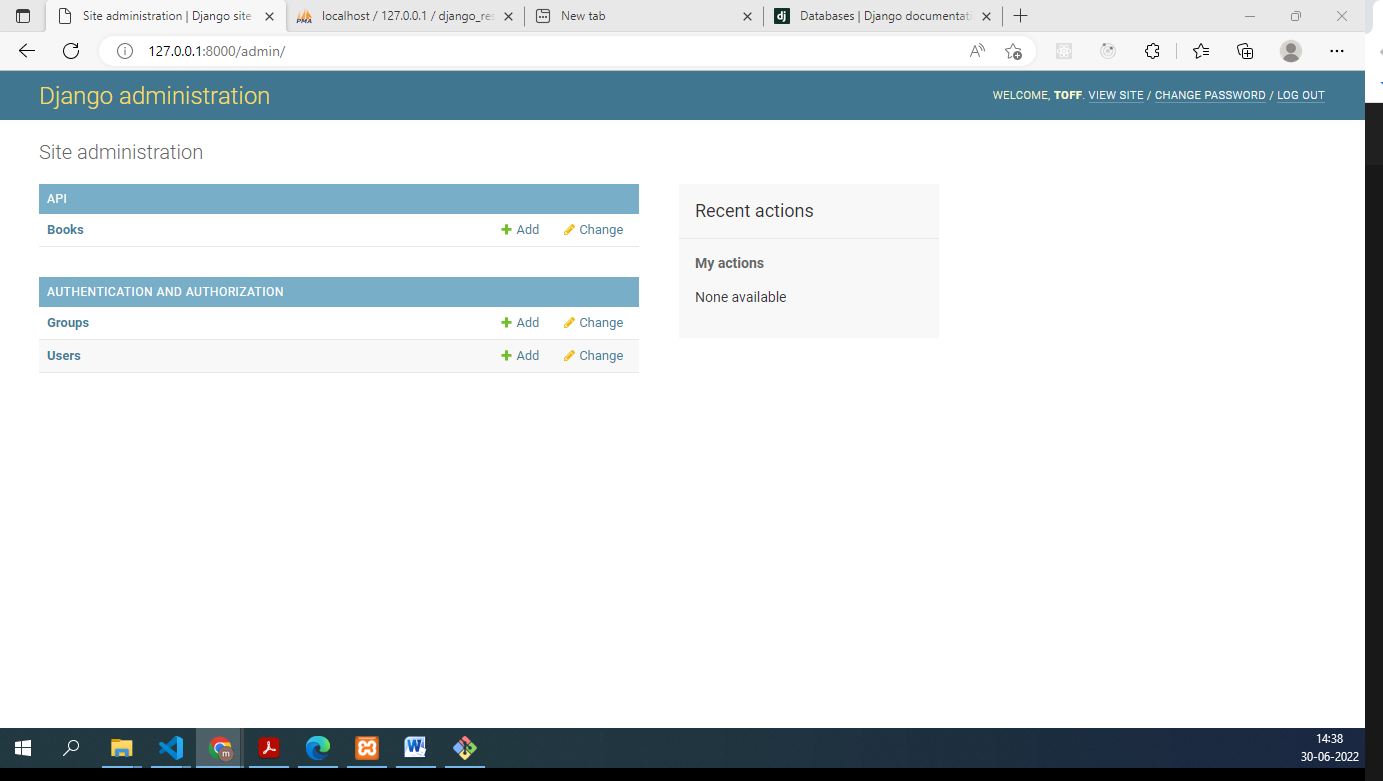


python manage.py createsuperuser

Username (leave blank to use 'hp'): toff

Email address: [toffcodes07@gmail.com](mailto:toffcodes07@gmail.com)

Password:123toff#



Added a author field later

Create A serializer

Serializers.py

from dataclasses import field

from rest\_framework import serializers

from .models import Book

class BookSerializer(serializers.ModelSerializer):

    class Meta:

        model = Book

        # field = ('title', 'author')

        fields = '\_\_all\_\_'

Creating All CRUD Operations

Views.py

from django.shortcuts import render

from rest\_framework.response import Response

from rest\_framework.decorators import api\_view

from .serializers import BookSerializer

from .models import Book

# Create your views here.

@api\_view(['GET'])

def apiOverview(request):

    api\_urls = {

        'List': '/book-list/',

        'Detail-view': '/book-detail/<int:id>',

        'Create': '/book-create/',

        'Update-view': '/book-update/<int:id>',

        'Delete-view': '/book-delete/<int:id>',

    }

    return Response(api\_urls)

@api\_view(['GET'])

def showAll(request):

    books = Book.objects.all()

    serializer = BookSerializer(books, many=True)

    return Response(serializer.data)

@api\_view(['GET'])

def viewBook(request, pk):

    books = Book.objects.get(id=pk)

    serializer = BookSerializer(books, many=False)

    return Response(serializer.data)

@api\_view(['POST'])

def createBook(request):

    serializer = BookSerializer(data=request.data)

    if(serializer.is\_valid()):

        serializer.save()

    return Response(serializer.data)

@api\_view(['POST'])

def updateBook(request, pk):

    books = Book.objects.get(id=pk)

    serializer = BookSerializer(instance=books, data=request.data)

    if(serializer.is\_valid()):

        serializer.save()

    return Response(serializer.data)

@api\_view(['GET'])

def deleteBook(request, pk):

    book = Book.objects.get(id=pk)

    book.delete()

    return Response("Item deleted Successfully")

urls.py

from django.urls import path

from api import views

urlpatterns = [

    path('', views.apiOverview, name='apiOverview'),

    path('book-list/', views.showAll, name='book-list'),

    path('book-detail/<int:pk>', views.viewBook, name='book-detail'),

    path('book-create/', views.createBook, name='book-create'),

    path('book-update/<int:pk>', views.updateBook, name='book-update'),

    path('book-delete/<int:pk>', views.deleteBook, name='book-delete'),

]