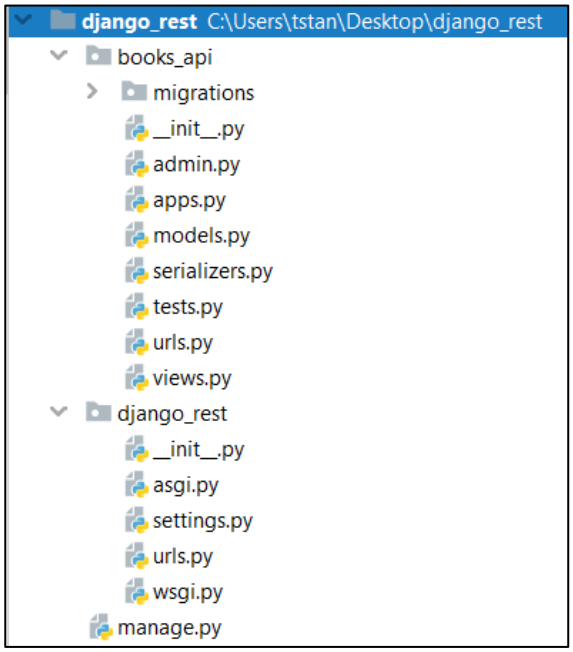
# Lab: Django REST Framework

## Books API

Create a simple **Books API** using the **Django REST framework**.

## 1. Create the Project

- **Create** your **project** and your **API app**




## 2. Setup

- Make sure that you have **`djangorestframework`** installed (if not, install it using pip)
- Add **`'rest_framework'`** to the **`INSTALLED_APPS`**
- Add **`'books_api'`** to the **`INSTALLED_APPS`**

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'books_api'
]
```

- **Configure** and **create** your **database**

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'booksdb',
        'USER': 'postgres',
        'PASSWORD': 'admin'
    }
}
```

## 3. Create your Model

Create the **Book model** as shown below

```
from django.db import models


# Create your models here.
class Book(models.Model):
    title = models.CharField(max_length=20)
    pages = models.IntegerField(default=0)
    description = models.TextField(max_length=100, default="")
    author = models.CharField(max_length=20)
```

Make **migrations** and **register the model** in the **`admin.py`** file

## 4. Create the Serializer

Create a new file called **`'serializers.py'`** and create the **`BookSerializer`**

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


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

## 5. Creating the Views

First, create the **`ListBookView`**

```python
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status
from .models import Book
from .serializers import BookSerializer


class ListBooksView(APIView):
    def get(self, req):
        books = Book.objects.all()
        serializer = BookSerializer(books, many=True)
        return Response({"books": serializer.data})

    def post(self, req):
        serializer = BookSerializer(data=req.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

And then, the **DetailBookView**

```python
class DetailBookView(APIView):
    def get(self, req, id):
        book = Book.objects.get(pk=id)
        serializer = BookSerializer(book)
        return Response({"book": serializer.data})

    def post(self, req, id):
        book = Book.objects.get(pk=id)
        serializer = BookSerializer(book, data=req.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.date, status=status.HTTP_200_OK)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

    def delete(self, req, id):
        book = Book.objects.get(pk=id)
        book.delete()
        return Response(status=status.HTTP_200_OK)
```

## 6. Configure the urls.py Files

Create **`urls.py`** file in the **app**, and add the **`urls`**

```
from django.urls import path
from . import views


urlpatterns = [
    path('books/', views.ListBooksView.as_view()),
    path('books/<int:id>', views.DetailBookView.as_view())
]
```

And the **`urls.py`** file in the **project**

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


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

## 7. Create a superuser

Create a **superuser** and **create some books**

## 8. Test The API