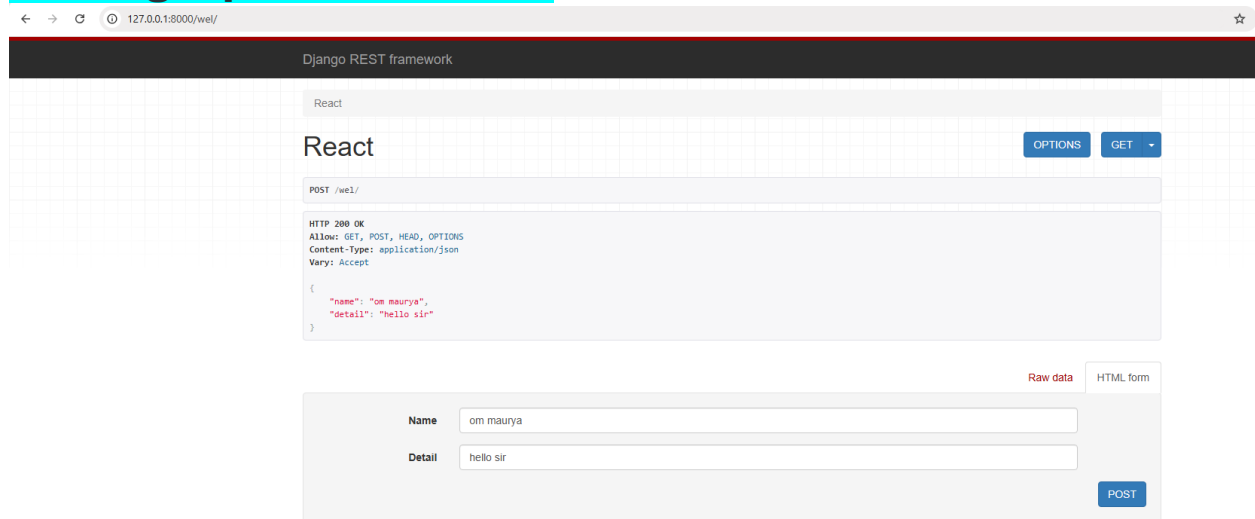


Setting Up the Backend



Create a project folder for Django by creating a virtual environment. You must have installed **virtualenv** package.

Step 1: Install Python

If not installed install it by typing a command on terminal.

```
python3 -m pip install --user virtualenv
```

Step 2: Create a virtual environment

Use this command to create a virtual environment

```
python3 -m venv env
cd env
source bin/activate
```

Step 3: Install Required Packages

Install below packages by using pip

```
pip install django
pip install djangorestframework
python -m pip install django-cors-headers
```

As shown below

```
C:\Users\Big Data>pip install djangorestframework
Collecting djangorestframework
  Downloading djangorestframework-3.15.2-py3-none-any.whl.metadata (10 kB)
Collecting django>=4.2 (from djangorestframework)
  Downloading Django-5.1.5-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.8.1 (from django>=4.2->djangorestframework)
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django>=4.2->djangorestframework)
  Downloading sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: tzdata in c:\users\big data\appdata\local\programs\python\python311\l
Downloaded djangorestframework-3.15.2-py3-none-any.whl (1.1 MB)
----- 1.1/1.1 MB 6.5 MB/s eta 0:00:00
Downloading Django-5.1.5-py3-none-any.whl (8.3 MB)
----- 8.3/8.3 MB 9.9 MB/s eta 0:00:00
Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Downloading sqlparse-0.5.3-py3-none-any.whl (44 kB)
Installing collected packages: sqlparse, asgiref, django, djangorestframework
Successfully installed asgiref-3.8.1 django-5.1.5 djangorestframework-3.15.2 sqlparse-0.5.3
```

```
C:\Users\Big Data>python -m pip install django-cors-headers
Collecting django-cors-headers
  Downloading django_cors_headers-4.6.0-py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: asgiref>=3.6 in c:\users\big data\appdata\local\programs\python\pytho
Requirement already satisfied: django>=4.2 in c:\users\big data\appdata\local\programs\python\pytho
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\big data\appdata\local\programs\python\p
Requirement already satisfied: tzdata in c:\users\big data\appdata\local\programs\python\python311\
Downloaded django_cors_headers-4.6.0-py3-none-any.whl (12 kB)
Installing collected packages: django-cors-headers
Successfully installed django-cors-headers-4.6.0
```

Step 4: initialize Project

Create a project name of your choice.

```
C:\Users\Big Data>d:
D:\>django-admin startproject quotes
```

After project folder created

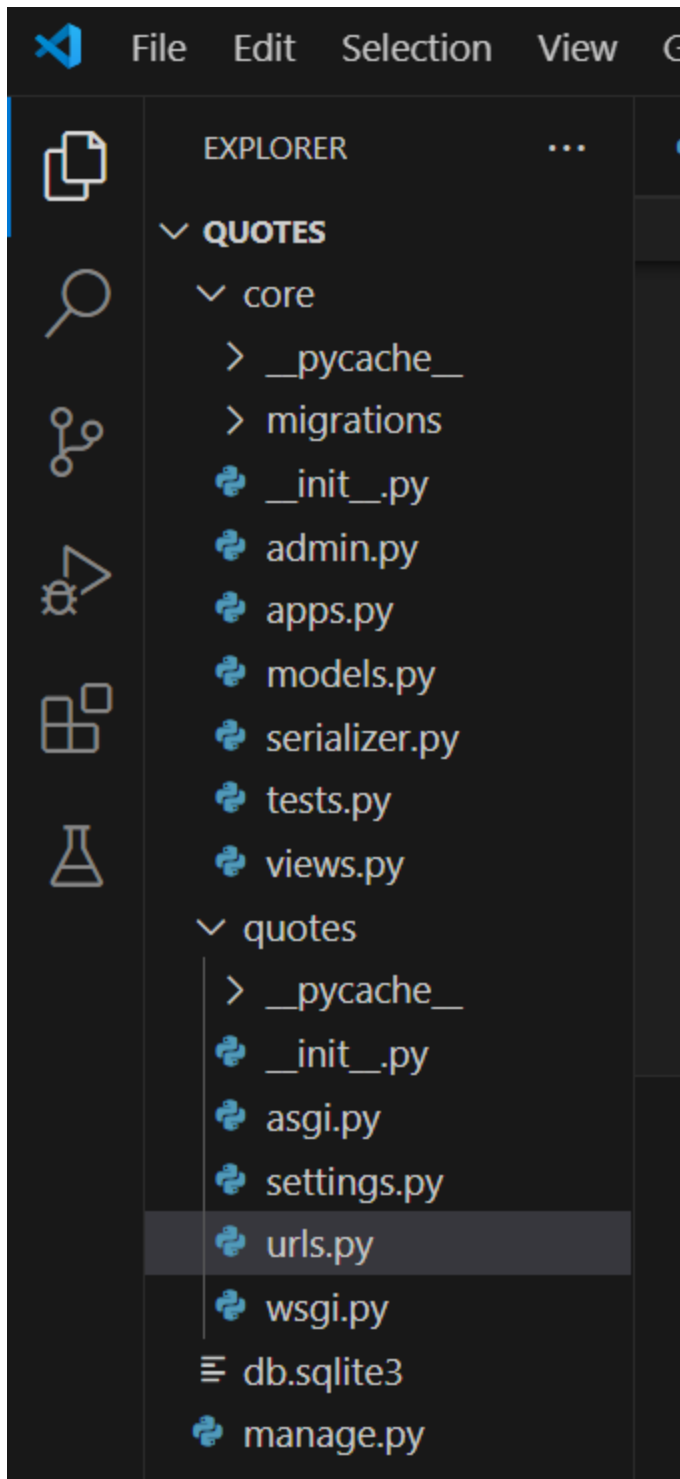
```
D:\>cd quotes
D:\quotes>
```

Create app

```
D:\quotes>django-admin startapp core
```

```
PS D:\quotes> python manage.py makemigrations
Migrations for 'core':
  core\migrations\0001_initial.py
    + Create model React
PS D:\quotes> █
```

Project folder Structure :-



`models.py` (inside core app folder):-

```
from django.db import models

# Create your models here.

class React(models.Model):
    name = models.CharField(max_length=30)
    detail = models.CharField(max_length=500)
```

`serializers.py` (inside core app folder):-

```
from rest_framework import serializers
from . models import *

class ReactSerializer(serializers.ModelSerializer):
    class Meta:
        model = React
        fields = ['name', 'detail']
```

`views.py` file(inside core app folder):-

```
from django.shortcuts import render

from rest_framework.views import APIView
from . models import *
from rest_framework.response import Response
from . serializer import *
# Create your views here.

class ReactView(APIView):

    serializer_class = ReactSerializer

    def get(self, request):
        detail = [ {"name": detail.name, "detail": detail.detail}
                  for detail in React.objects.all()]
        return Response(detail)

    def post(self, request):
```

```
serializer = ReactSerializer(data=request.data)
if serializer.is_valid(raise_exception=True):
    serializer.save()
    return Response(serializer.data)
```

`settings.py` (inside quotes project folder)

Update Configuration

There are few changes in `settings.py` file which are listed below

1. Add `rest_framework`, `core`, `corsheaders` to INSTALLED APPS

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

2. Add `corsheaders.middleware.CorsMiddleware` to MIDDLEWARE list

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'corsheaders.middleware.CorsMiddleware',
]
```

Now let's get back to the final part of our back-end. Run the following commands on the terminal.

- This command is used to detect changes made in the database.

```
python manage.py makemigrations
```

- This command applies those changes to the database.

```
python manage.py migrate
```

- For creating a superuser who will be the admin of the whole app.

```
python manage.py createsuperuser --email admin@example.com --username admin
```

- This command will run the server and the server should always be in running state.

```
python manage.py runserver
```

as shown below examples:-

```
PS D:\quotes> python manage.py makemigrations
Migrations for 'core':
  core\migrations\0001_initial.py
    + Create model React
PS D:\quotes> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, core, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
```

```
PS D:\quotes> python manage.py runserver
>>>
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 15, 2025 - 12:38:20
Django version 5.1.5, using settings 'quotes.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Finally output:-

127.0.0.1:8000/wel/

Django REST framework

React

React OPTIONS GET

POST /wel/

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

```
{
  "name": "om maurya",
  "detail": "hello sir"
}
```

[Raw data](#) [HTML form](#)

Name

Detail

POST