**Setup Django REST Framework**

[[Piyush](https://piyushthedev.medium.com/?source=post_page-----e839e8b287c7--------------------------------)](https://piyushthedev.medium.com/?source=post_page-----e839e8b287c7--------------------------------)

[Piyush](https://piyushthedev.medium.com/?source=post_page-----e839e8b287c7--------------------------------)

·

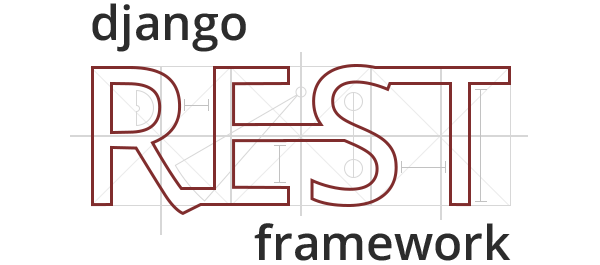
[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fda9393796e82&operation=register&redirect=https%3A%2F%2Fpiyushthedev.medium.com%2Fsetup-django-rest-framework-e839e8b287c7&user=Piyush&userId=da9393796e82&source=post_page-da9393796e82----e839e8b287c7---------------------post_header-----------)

4 min read

·

3 days ago

4



Setting up the Django REST Framework is sometimes very confusing, so here’s how to do it the smarter way.

*Pre-requisites*

1. You must have Python installed in your system and Windows user must have Python added to their environment variables. To verify installation just type python --version in your terminal, and you will get the version of Python installed in your system.
2. You can install Python from [here](https://www.python.org/).

* For **Linux users**, run python3 --version in your terminal to get the version of Python3.

Now, let’s move on to the setup part.

Fire up your terminal, and run the following commands:

**#1 — Installing Django library from pip.**

* For Windows users:

pip install django

* For Linux users:

sudo python3 -m pip install django

This will install the Django library in your system.

**#2 — Now let’s create a Django project.**

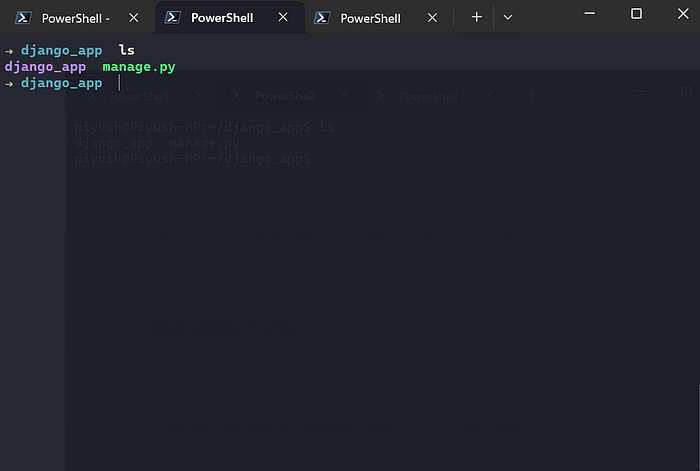
* For Windows users:

python -m django startproject <your\_project>

* For Linux users:

python3 -m django startproject <your\_project>

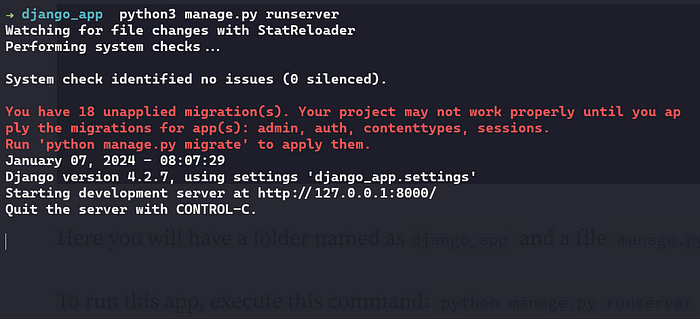
Now when you run cd <your\_project> you will see something like this:



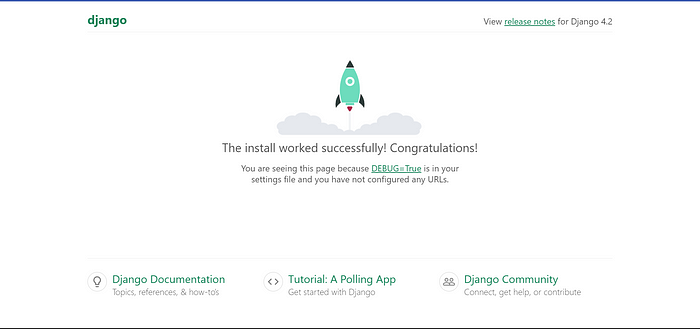
Here you will have a folder named as <your\_project> and a file manage.py

To run this app, execute this command: python manage.py runserver or python3 manage.py runserver (for **Linux users**).

You will see something like this in your terminal:



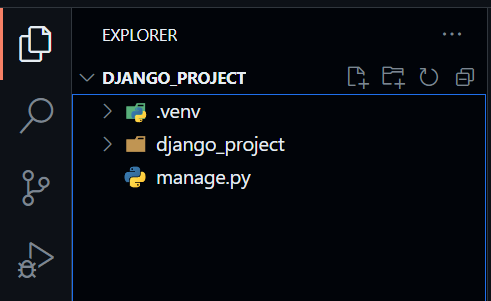
When you go to this URL http://localhost:8000 you will see a beautiful interface like this:



**#3 — The Python Virtual Environment**

* From this step, you can follow two approaches, either you can create a Python virtual environment to keep all the dependencies separate from your system, or you can directly install all the dependencies on your system itself. The latter is not so good approach.
* To create a virtual environment in Windows, run python -m venv .venv inside your project’s root directory
* In Linux, you have to run python3 -m venv .venv

This will create a .venv folder inside your project’s root directory like this:

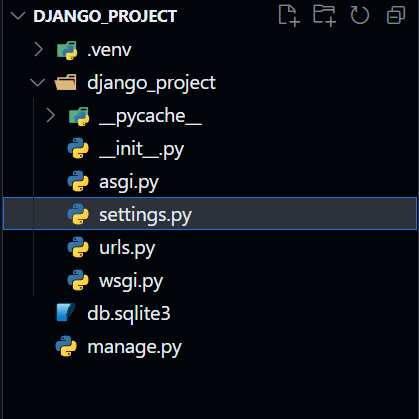


* To activate this environment, run .venv/Scripts/activate in your Windows terminal.
* Linux users, run source .venv/bin/activate
* Now your Python virtual environment is activated, from now on all the libraries will be installed in this environment only. So we have to install Django in this environment also. Just execute the above-mentioned commands for the same.

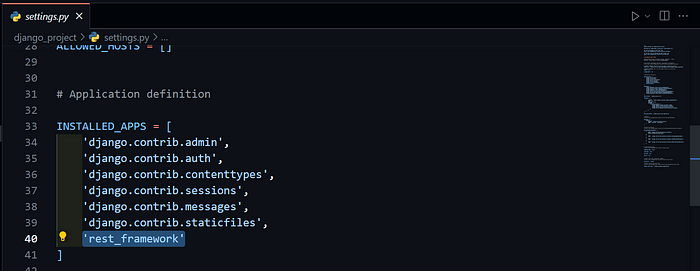
**#4 — The Django REST Framework**

The Django REST framework allows you to create a REST API on top of your Django app, so let’s see how to install that.

* To install it, run pip install djangorestframework
* Next, open settings.py inside the folder, it will be present inside your project folder like this:



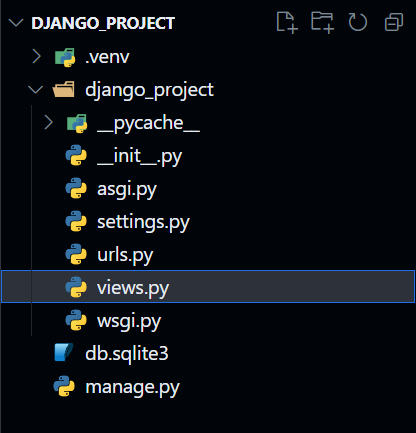
* Add rest\_framework inside the INSTALLED\_APPS list like this:



* Inside this list, you can also add your Django apps, to create a Django for for project, run python -m django startapp example
* It will create a folder example inside your project’s root directory.
* In this tutorial, I will use the given app only, that is the django\_project folder in my case. All the steps for any other app will be the same.

**#4 — Inside Django App**

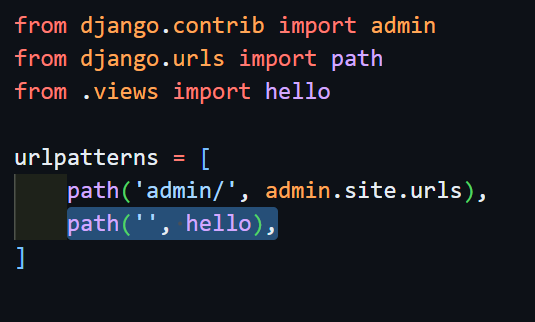
* Create a file views.py inside <your\_project> a directory like this:



* Let’s create a sample view inside the file views.py

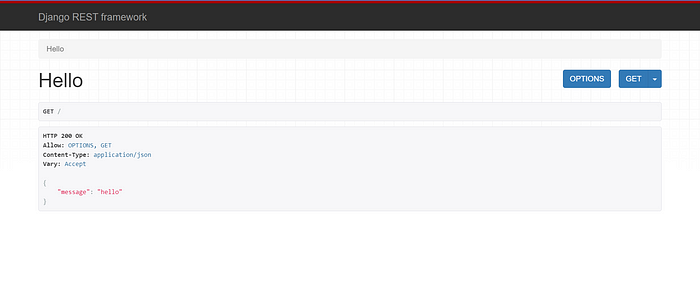
from rest\_framework.response import Response  
from rest\_framework.decorators import api\_view  
  
  
@api\_view(['GET'])  
def hello(request):  
 return Response({'message': 'hello'})

* Above is the simple hello function which returns a Response object.
* The decorator @api\_view is used to specify the HTTP methods that the function will allow.
* The next step is to create a route for the view just created. Open the file urls.py file inside your <your\_project> directory and write like this:



* As you can see I have imported the hello function from the same module in which the urls.py file is there. So if you have to import from some other module(other Django app), you have to write from app\_name.views import function\_name

**#5 — The Output**



* You see we have received our response🤩.
* Similarly, you can create the same for POST requests.

All the best guys!

If you come across any issues while setting up the app and need some help, don’t hesitate to contact me at — **piyush.at.googl@gmail.com**.

**Give a clap if you like it😊.**

[Django](https://medium.com/tag/django?source=post_page-----e839e8b287c7---------------django-----------------)

[Django Rest Framework](https://medium.com/tag/django-rest-framework?source=post_page-----e839e8b287c7---------------django_rest_framework-----------------)

[Backend](https://medium.com/tag/backend?source=post_page-----e839e8b287c7---------------backend-----------------)

[API](https://medium.com/tag/api?source=post_page-----e839e8b287c7---------------api-----------------)

[Python](https://medium.com/tag/python?source=post_page-----e839e8b287c7---------------python-----------------)