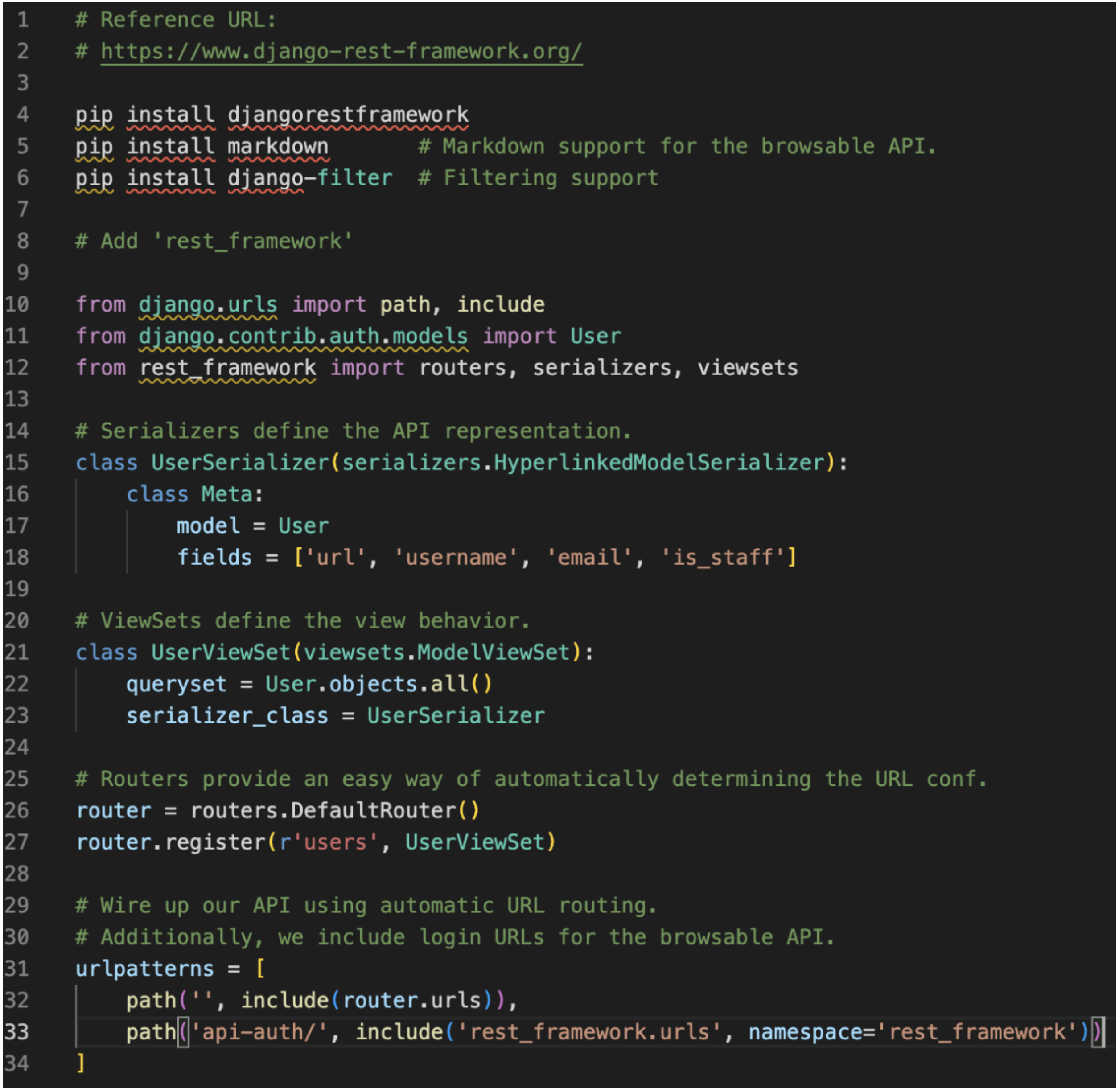
Original script:

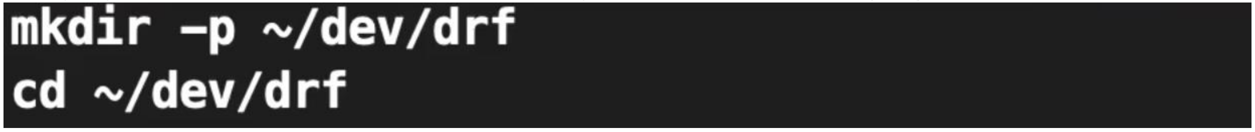


```
# Reference URL:
# https://www.django-rest-framework.org/

pip install djangorestframework
pip install markdown       # Markdown support for the browsable API.
pip install django-filter  # Filtering support

# Add 'rest_framework'

from django.urls import path, include
from django.contrib.auth.models import User
from rest_framework import routers, serializers, viewsets

# Serializers define the API representation.
class UserSerializer(serializers.HyperlinkedModelSerializer):
    class Meta:
        model = User
        fields = ['url', 'username', 'email', 'is_staff']

# ViewSets define the view behavior.
class UserViewSet(viewsets.ModelViewSet):
    queryset = User.objects.all()
    serializer_class = UserSerializer

# Routers provide an easy way of automatically determining the URL conf.
router = routers.DefaultRouter()
router.register(r'users', UserViewSet)

# Wire up our API using automatic URL routing.
# Additionally, we include login URLs for the browsable API.
urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls', namespace='rest_framework'))
]
```

I was trying to build a Web API using Django REST framework using Python.

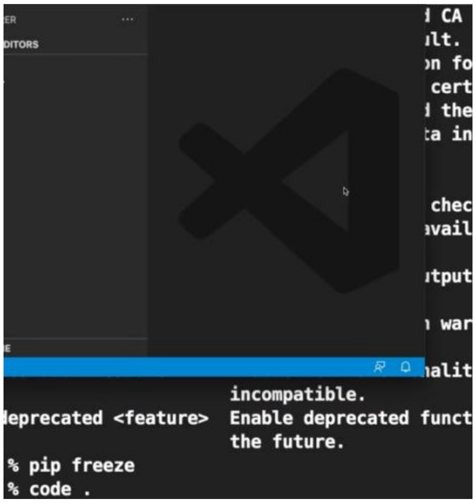
# To test it's working: python-request

1. Setup Python virtual environment, install requirements and start Django



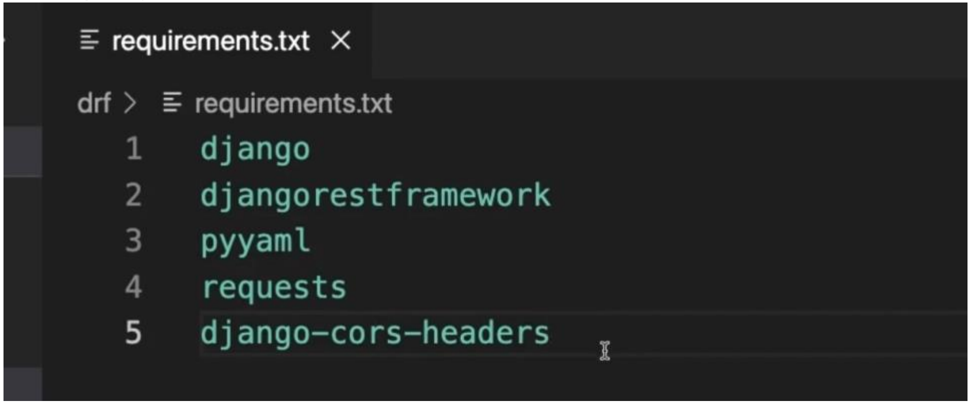
```
mkdir -p ~/dev/drf
cd ~/dev/drf
```

2. A cool way to open VS code:



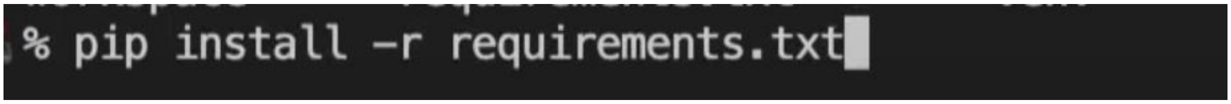
```
ı CA
ılt.
on fo
 cert
ı the
ta in

 chec
avail

ıtput

ı war

nalit
                     incompatible.
deprecated <feature>  Enable deprecated funct
                     the future.
% pip freeze
% code .
```

3. Adding requirenments.txt



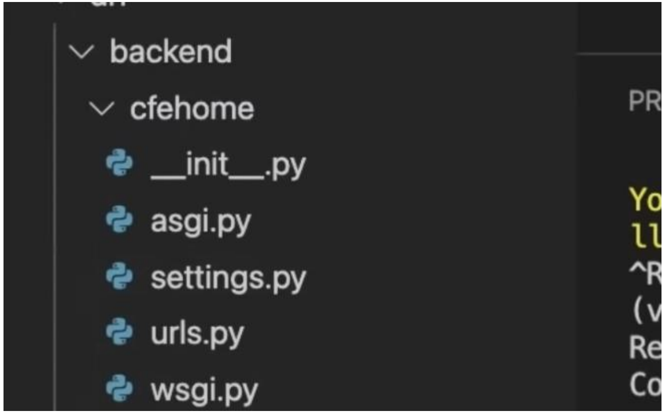
```
django
djangorestframework
pyyaml
requests
django-cors-headers
```

4. RUN IT



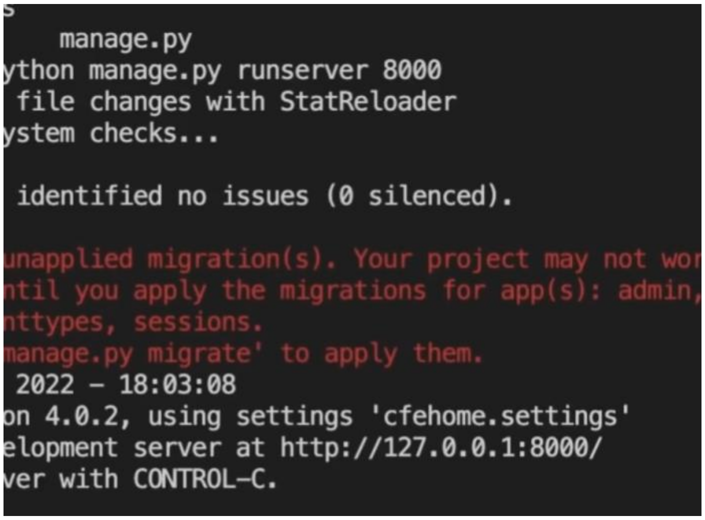
```
% pip install -r requirements.txt
```

5. Backend: where Django framework resides; and we call project cfehome

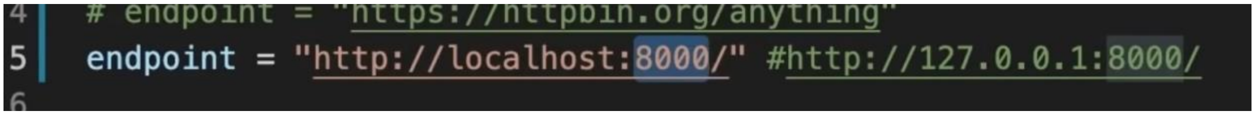


```
backend
  cfehome
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
```

6. Crete a python client & setup endpint

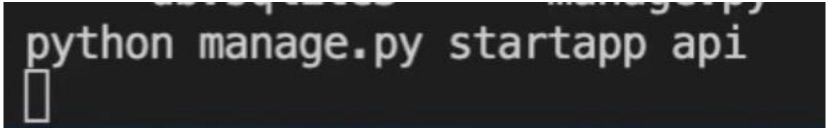


```
s
    manage.py
ython manage.py runserver 8000
 file changes with StatReloader
ystem checks...

 identified no issues (0 silenced).

unapplied migration(s). Your project may not wor
ntil you apply the migrations for app(s): admin,
nttypes, sessions.
manage.py migrate' to apply them.
 2022 - 18:03:08
on 4.0.2, using settings 'cfehome.settings'
elopment server at http://127.0.0.1:8000/
ver with CONTROL-C.
```



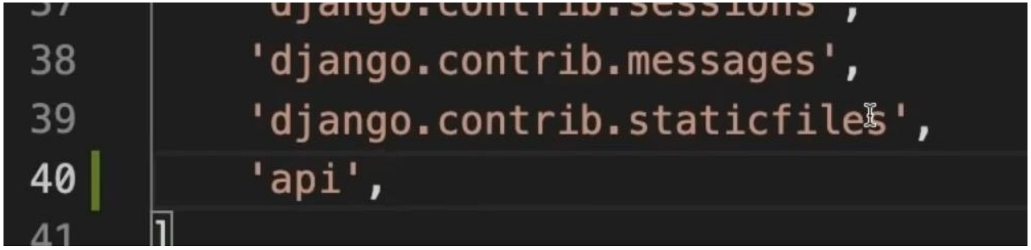
```
# endpoint = "https://httpbin.org/anything"
endpoint = "http://localhost:8000/" #http://127.0.0.1:8000/
```

7. Create app called api



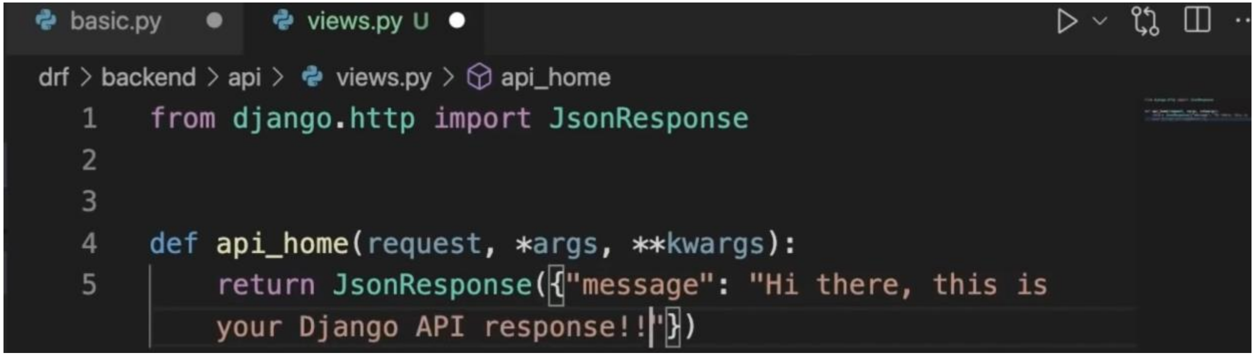
```
python manage.py startapp api
```

Also inside settings.py:



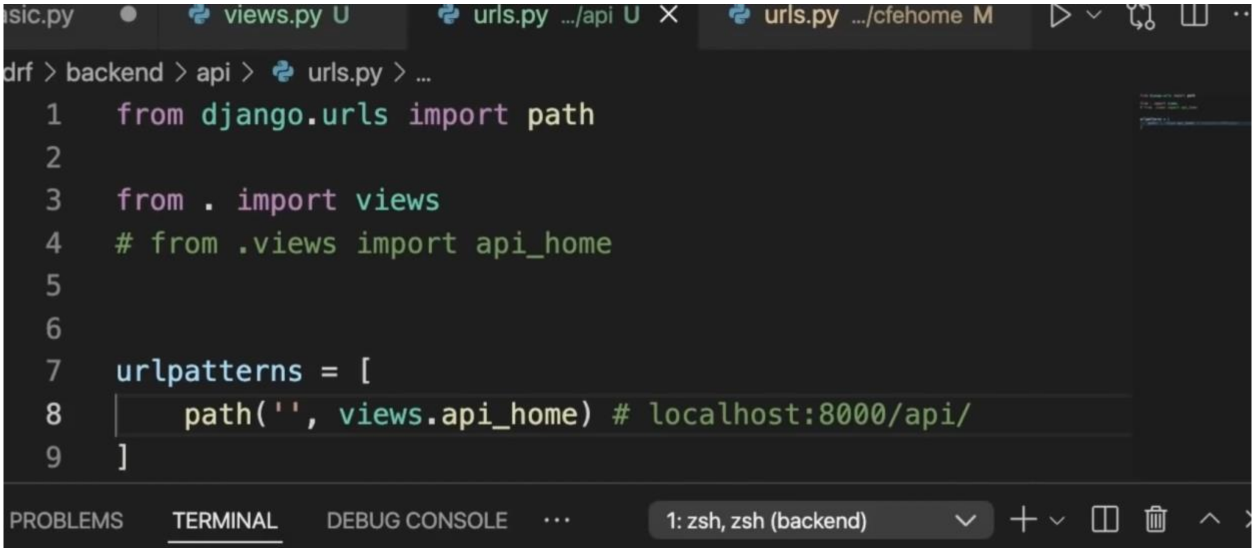
```
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'api',
]
```

8. Create view.py



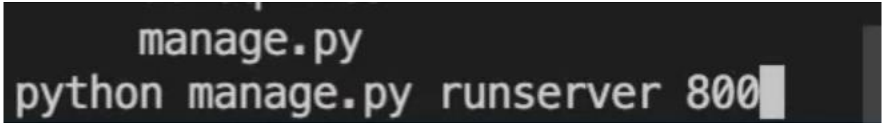
```
from django.http import JsonResponse


def api_home(request, *args, **kwargs):
    return JsonResponse({"message": "Hi there, this is
    your Django API response!!"})
```

9. Inside urls.py: specify path



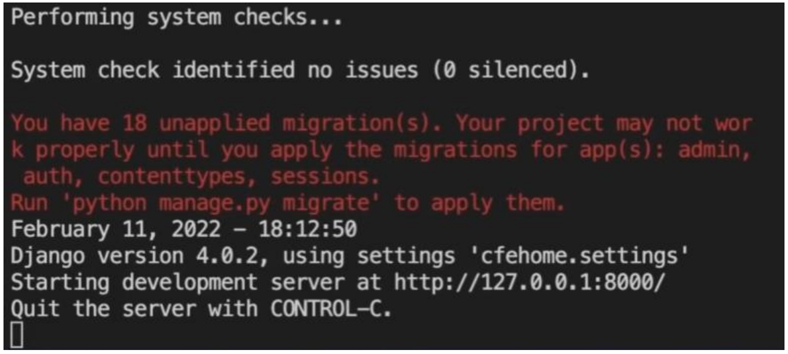
```python
from django.urls import path

from . import views
# from .views import api_home


urlpatterns = [
    path('', views.api_home) # localhost:8000/api/
]
```

10. Run our project:



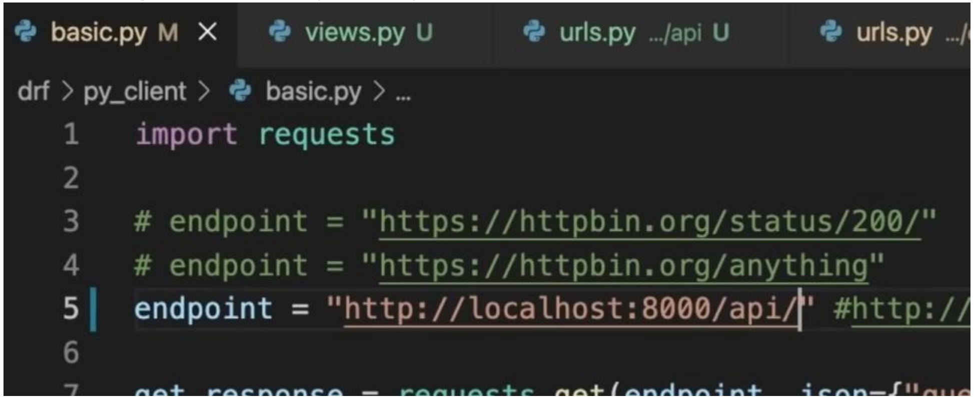
```
    manage.py
python manage.py runserver 800
```

Got the port going



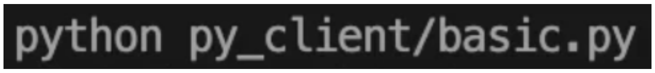
```
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not wor
k properly until you apply the migrations for app(s): admin,
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 11, 2022 - 18:12:50
Django version 4.0.2, using settings 'cfehome.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

And adding our new url path /api



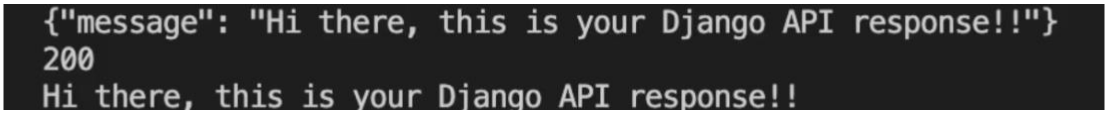
```python
import requests

# endpoint = "https://httpbin.org/status/200/"
# endpoint = "https://httpbin.org/anything"
endpoint = "http://localhost:8000/api/" #http://

get_response = requests.get(endpoint, json={"que
```

After run



```
python py_client/basic.py
```

We can get



```
{"message": "Hi there, this is your Django API response!!"}
200
Hi there, this is your Django API response!!
```