

# REST API

---

## 1. 가상환경 설치.

```
python -m venv myenv
```

django를 설치하기 전에 가상환경 설치.

그리고, 가상환경을 실행.

```
. myenv/Scripts/activate
```

또는,

```
source myenv/Scripts/activate
```

가상환경 실행 후 django와 restframework 설치.

```
pip install django  
pip install djangorestframework
```

## 2. 장고 프로젝트&앱 생성

```
django-admin startproject proj  
cd proj/  
python manage.py startapp mystorage
```

프로젝트를 만든 후, 프로젝트 폴더에 들어가 app까지 생성.

## 3. settings.py에 앱들 추가.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    ## 새로운 앱 추가  
    'rest_framework',  
    'mystorage'  
]
```

## 4. mystorage 앱에 urls.py파일 추가.

mystorage에 urls.py파일 추가 후, proj폴더 안에 urls.py에 추가.

```

from django.contrib import admin
# include 추가
from django.urls import path, include
# mystorage.urls 추가
import mystorage.urls

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('mystorage.urls'))
]

```

mystorage의 urls.py에 router로 urls 설정.

```

from rest_framework.routers import DefaultRouter
from django.urls import path, include
from . import views

router = DefaultRouter()

router.register('essay', views.PostViewSet),

urlpatterns = [
    path('', include(router.urls))
]

```

## 5. mystorage의 views파일에 viewset 추가.

```

from django.shortcuts import render
## viewset과 model 불러오기
from rest_framework import viewsets
from .models import Essay
# Create your views here.
class PostViewSet(viewsets.ModelViewSet):
    queryset = Essay.objects.all()

```

## 6. Essay 모델 추가.

```

from django.db import models
## AUTH_USER_MODEL을 불러오기 위한 settings import
from django.conf import settings

# Create your models here.
class Essay(models.Model):
    author = models.ForeignKey(
        settings.AUTH_USER_MODEL, default=1, on_delete=models.CASCADE)
    title = models.CharField(max_length=30)
    body = models.TextField()

```

model을 추가 했으니,

```

python manage.py makemigrations
python manage.py migrate

```

## 7. mystorage폴더에 serializers.py 추가

```

from rest_framework import serializers
from .models import Essay

class EssaySerializer(serializers.ModelSerializer):
    class Meta:
        model = Essay
        fields = "__all__"

```

mystorage폴더 views.py에 EssaySerializer 추가.

```

from django.shortcuts import render
from rest_framework import viewsets
from .models import Essay
# serializer 추가.
from .serializers import EssaySerializer

# Create your views here.
class PostViewSet(viewsets.ModelViewSet):
    queryset = Essay.objects.all()
    # serializer 추가.
    serializer_class = EssaySerializer

```

## 8. 실행시켜보기.

```
python manage.py runserver
```

하지만, 글을 못쓰 -> author에 유저를 꼭 넣어줘야하기 때문.

```
python manage.py createsuperuser
```

admin을 만들고 admin 페이지에서 유저 생성 후 글쓰기 가능.

그리고, author를 username으로 쓰고 싶으니까 Serializer를 수정.

```

class EssaySerializer(serializers.ModelSerializer):
    author_name = serializers.ReadOnlyField(source='author.username')

    class Meta:
        model = Essay
        ## 수정
        fields = (
            'pk', 'title', 'body', 'author_name'
        )

```

그리고, PostViewSet도 수정.

```

class PostViewSet(viewsets.ModelViewSet):
    queryset = Essay.objects.all()
    serializer_class = EssaySerializer
    ## 수정
    def perform_create(self, serializer):
        serializer.save(author=self.request.user)

```

좀 더 로그인을 편하게 하기위해서 proj폴더 urls파일에 api-auth 추가.

```

from django.contrib import admin
from django.urls import path, include
import mystorage.urls
## api-auth 추가
import rest_framework.urls

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('mystorage.urls')),
    #api-auth 추가
    path('api-auth/', include('rest_framework.urls'))
]

```

## 9. 해당 유저가 쓴 글만 보이게 하기.

views.py에 PostViewSet에 get\_queryset() 추가.

```

class PostViewSet(viewsets.ModelViewSet):
    queryset = Essay.objects.all()
    serializer_class = EssaySerializer

    def perform_create(self, serializer):
        serializer.save(author=self.request.user)
    # queryset 추가
    def get_queryset(self):
        qs = super().get_queryset()

        if self.request.user.is_authenticated:
            qs = qs.filter(author=self.request.user)
        else:
            qs = qs.none()
        return qs

```

검색창 만들어 보기.

```

from django.shortcuts import render
from rest_framework import viewsets
from .models import Essay
from .serializers import EssaySerializer
## SearchFilter 추가
from rest_framework.filters import SearchFilter

class PostViewSet(viewsets.ModelViewSet):
    queryset = Essay.objects.all()
    serializer_class = EssaySerializer
    #SearchFilter 설정
    filter_backends = [SearchFilter]
    search_fields = ('title', 'body')

```

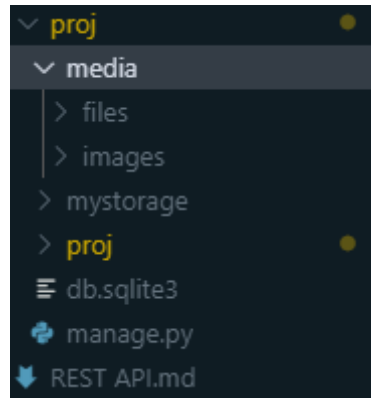
## 10. media파일 설정.

settings.py에 media를 위한 설정 추가.

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/

STATIC_URL = '/static/'

## media를 위한 설정
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```



media폴더와 media폴더 안에 files, images폴더 생성  
media파일에 대한 urls을 proj폴더 urls.py에 설정.

```
from django.contrib import admin
from django.urls import path, include
import mystorage.urls
import rest_framework.urls
## media 파일 위치 추가를 위한 import
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('mystorage.urls')),
    path('api-auth/', include('rest_framework.urls'))
]

## media 파일 위치 추가.
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

## 11. model에 Album과 Files 추가!

```
# Create your models here.
class Essay(models.Model):
    author = models.ForeignKey(
        settings.AUTH_USER_MODEL, default=1, on_delete=models.CASCADE)
    title = models.CharField(max_length=30)
    body = models.TextField()

## Album , Files 추가.
class Album(models.Model):
    author = models.ForeignKey(
        settings.AUTH_USER_MODEL, default=1, on_delete=models.CASCADE)
    image = models.ImageField(upload_to="images")
    desc = models.CharField(max_length=100)
```

```
class Files(models.Model):
    author = models.ForeignKey(
        settings.AUTH_USER_MODEL, default=1, on_delete=models.CASCADE)
    myfile = models.FileField(blank=False,
                              null=False, upload_to="files")
    desc = models.CharField(max_length=100)
```

@여기서 주의. 이미지 파일을 위해서는 pillow 필요.

```
pip install pillow
```

pillow 라이브러리 설치.

```
python manage.py makemigrations
python manage.py migrate
```

모델은 추가 후 migrate 꼭 실행.

## 12. mystorage폴더의 urls 추가, views.py viewset 추가.

urls.py

```
router = DefaultRouter()

router.register('essay', views.PostViewSet)
## router에 추가.
router.register('Album', views.ImageViewSet)
router.register('Files', views.FileViewSet)
```

views.py

```
from django.shortcuts import render
from rest_framework import viewsets
## Album, Files 모델 추가.
from .models import Essay, Album, Files
## serializer 추가.
from .serializers import EssaySerializer, AlbumSerializer, FileSerializer
from rest_framework.filters import SearchFilter

class PostViewSet(viewsets.ModelViewSet):
    queryset = Essay.objects.all()
    serializer_class = EssaySerializer

    filter_backends = [SearchFilter]
    search_fields = ('title', 'body')

    def perform_create(self, serializer):
        serializer.save(author=self.request.user)

    def get_queryset(self):
        qs = super().get_queryset()

        if self.request.user.is_authenticated:
            qs = qs.filter(author=self.request.user)
        else:
```

```

        qs = qs.none()
    return qs

## viewset들 추가.
class ImageViewSet(viewsets.ModelViewSet):
    queryset = Album.objects.all()
    serializer_class = AlbumSerializer

class FileViewSet(viewsets.ModelViewSet):
    queryset = Files.objects.all()
    serializer_class = FileSerializer

```

### 13. serializer 생성.

```

from rest_framework import serializers
# model들 추가.
from .models import Essay, Album, Files

class EssaySerializer(serializers.ModelSerializer):
    author_name = serializers.ReadOnlyField(source='author.username')

    class Meta:
        model = Essay
        fields = (
            'pk', 'title', 'body', 'author_name'
        )

## Serializer 선언.
class AlbumSerializer(serializers.ModelSerializer):
    author_name = serializers.ReadOnlyField(source='author.username')
    image = serializers.ImageField(use_url=True)

    class Meta:
        model = Album
        fields = (
            'pk', 'author_name', 'image', 'desc'
        )

class FileSerializer(serializers.ModelSerializer):
    author = serializers.ReadOnlyField(source='author.username')
    myfile = serializers.FileField(use_url=True)

    class Meta:
        model = Album
        fields = (
            'pk', 'author', 'myfile', 'desc'
        )

```

여기까지 하면 다 된 것 같지만..file업로드가 에러가 남.

```

from django.shortcuts import render
## serializers, status 추가
from rest_framework import viewsets, serializers, status
from .models import Essay, Album, Files
from .serializers import EssaySerializer, AlbumSerializer, FileSerializer
from rest_framework.filters import SearchFilter

```

```

## Response와 Parser를 추가.
from rest_framework.parsers import MultiPartParser, FormParser
from rest_framework.response import Response

# Create your views here.
class PostViewSet(viewsets.ModelViewSet):
    queryset = Essay.objects.all()
    serializer_class = EssaySerializer

    filter_backends = [SearchFilter]
    search_fields = ('title', 'body')

    def perform_create(self, serializer):
        serializer.save(author=self.request.user)

    def get_queryset(self):
        qs = super().get_queryset()

        if self.request.user.is_authenticated:
            qs = qs.filter(author=self.request.user)
        else:
            qs = qs.none()
        return qs

class ImageViewSet(viewsets.ModelViewSet):
    queryset = Album.objects.all()
    serializer_class = AlbumSerializer

class FileViewSet(viewsets.ModelViewSet):
    queryset = Files.objects.all()
    serializer_class = FilesSerializer
    ## 파일 업로드를 위한 parser_classes와 post 오버라이딩.
    # parser_class
    parser_classes = (MultiPartParser, FormParser)

    # create()
    def post(self, request, *args, **kwargs):
        serializer = FilesSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        else:
            return Response(serializer.errors,
                            status=status.HTTP_400_BAD_REQUEST)

```

## 추가

```

REST_FRAMEWORK = {
    'DEFAULT_PAGINATION_CLASS':
    'rest_framework.pagination.LimitOffsetPagination',
    'PAGE_SIZE': 10,
}

```



페이지네이션.

```
from rest_framework.permissions import IsAuthenticated
from rest_framework.authentication import SessionAuthentication,
TokenAuthentication

authentication_classes = [TokenAuthentication, SessionAuthentication]
permission_classes = [IsAuthenticated]
```

auth and permission\

```
'rest_framework.authtoken',
```

토큰인 경우 app추가와 migrate 필요.

```
python manage.py drf_create_token 사용자명
```

토큰생성가능.