

Python Django

6강

여러분... 오늘이 마지막이에요ㅠㅠ

정말 다들 수고 많으셨습니다!!!

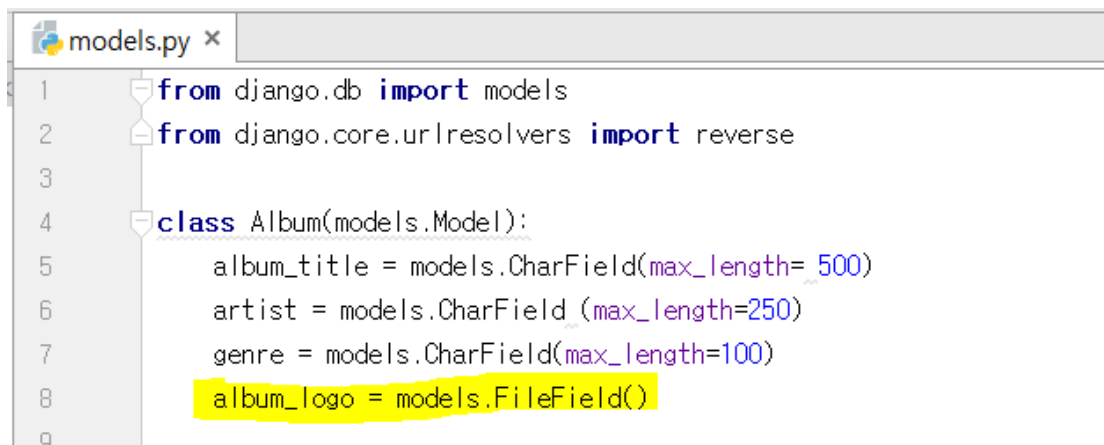
끝까지 같이해줘서 감사해요

Python Django

1. 파일 업로드 하기

여태까지는 앨범 로고의 url을 올리도록 했지만, 이제는 직접 사진을 업로드 할 수 있도록 만들어 볼게요!

1) Music의 models.py 의 album_logo 부분 수정
기존의 char -> file을 올릴 수 있도록 attribute 형식 수정



```
1 from django.db import models
2 from django.core.urlresolvers import reverse
3
4 class Album(models.Model):
5     album_title = models.CharField(max_length=500)
6     artist = models.CharField(max_length=250)
7     genre = models.CharField(max_length=100)
8     album_logo = models.FileField()
9
```

잠깐!
이거 하기 전에 DB에 있는
모든 데이터를 삭제해야 해요ㅠㅠ

2) Model을 수정했으니 DB 업데이트를 해줘야겠죠? cmd에서
 >Python manage.py makemigrations music
 >Python manage.py migrate

Python Django

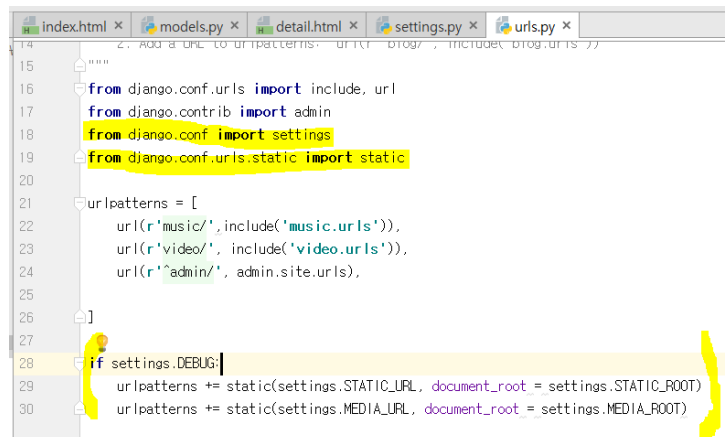
1. 파일 업로드 하기

3) 이제 사용자가 올린 파일을 어디다 저장해야 할지 그 길을 만들어 줘야 함!
Website의 setting.py 에 들어가서 아래 내용 추가

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')  
MEDIA_URL = '/media/'
```

-> 실제 컴퓨터에 저장될 디렉토리
-> 브라우저에서 http를 이용해서 접근할 수 있도록 하는 상대 url

4) 후 website의 urls.py에 추가



```
15  
16 from django.conf.urls import include, url  
17 from django.contrib import admin  
18 from django.conf import settings  
19 from django.conf.urls.static import static  
20  
21 urlpatterns = [  
22     url(r'music/', include('music.urls')),  
23     url(r'video/', include('video.urls')),  
24     url(r'admin/', admin.site.urls),  
25 ]  
26  
27  
28 if settings.DEBUG:  
29     urlpatterns += static(settings.STATIC_URL, document_root = settings.STATIC_ROOT)  
30     urlpatterns += static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
```

*Debug 가 true 면 development mode
나중에 실제로 서비스 할때는 false로 바꿔주면 된다

Python Django

1. 파일 업로드 하기

5)html에서 logo를 불러왔던 부분을 전부 고치기

`album.album_logo -> album.album_logo.url`

* 과거 우리가 넘겨줬던 것은 url이었지만
현재는 그냥 파일 그 자체로 보내주게 된다!
그렇기 때문에 .url을 붙여줘서 파일 위치를 전송할 수 있도록 해준다

Index.html에서

```
{% if all_albums %}
{% for album in all_albums %}
<div class="col-sm-4 col-lg-2">
  <div class="thumbnail">
    <a href="{% url 'music:detail' album.id %}">
      
    </a>
  </div>
</div>
```

Detail.html에서

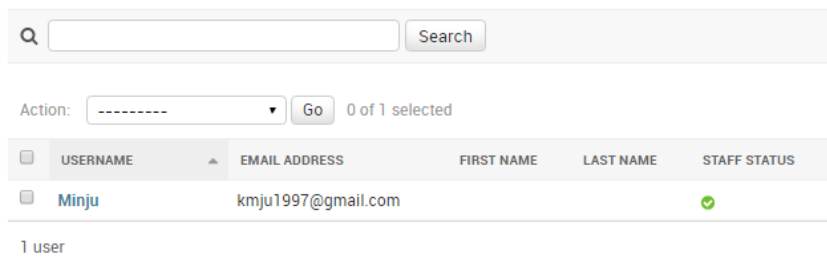
```
<!-- Left Album Info -->
<div class="col-sm-4 col-md-3">
  <div class="panel panel-default">
    <div class="panel-body">
      <a href="{% url 'music:detail' album.id %}">
        
      </a>
    </div>
  </div>
</div>
```

Python Django

2. User Registration

이제는 사이트 사용자가 회원가입을 하고 로그인을 할 수 있도록 하는 기능을 만들어 보도록 하겠습니다!

Select user to change

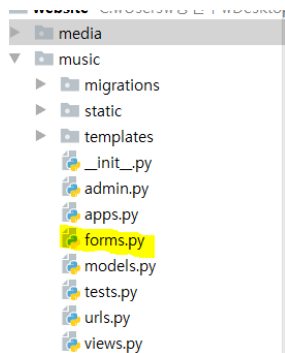


	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Minju	kmju1997@gmail.com			✓

1 user

실제로 우리의 user 내용은 admin 내용에 이미 존재하고 있다는 것!
이런 식으로 사용자들의 정보들이 여기에 저장 될거예요

1) Music에 forms.py 파일 새로 만들기



우리가 view의 createalbum 기능 처럼 user registration도 비슷하게 만들 거예요
이 forms.py는 가입할 때 내용을 넣는 form의 기초가 되는 것이죠

Python Django

2. User Registration

2) forms.py에 내용 채우기

```
1 from django.contrib.auth.models import User
2 from django import forms
3
4 class UserForm(forms.ModelForm):
5     password = forms.CharField(widget=forms.PasswordInput)
6
7     class Meta:
8         model = User
9         fields = ['username', 'email', 'password']
```

-> User model class 를 사용하기 위해서 import

-> password가 **이렇게 나타내도록 하기 위해서 저걸 작성해줌

-> Meta : class에 대한 정보를 나타냄

3) 회원가입 위한 view 만들기

작성 전에 import하기

```
1 from django.views import generic
2 from django.views.generic.edit import CreateView, UpdateView, DeleteView
3 from django.shortcuts import render, redirect
4 from django.contrib.auth import authenticate, login
5 from django.core.urlresolvers import reverse_lazy
6 from django.views.generic import View
7 from .models import Album
8 from .forms import UserForm
```

Python Django

2. User Registration

3) 회원가입 위한 view 만들기 - 아래와 같이 작성

```
class UserFormView(View):
    from_class = UserForm    # 어떤 기초 form을 쓸 건지 -> 방금 만들어준 form class를 적어주자
    template_name = 'music/registration_form.html'    # 사용할 html -> 후에 이어서 제작 할 것임

    #display blank form
    def get(self, request):
        form = self.form_class(None)
        return render(request, self.template_name, {'form': form})

    #process form data
    def post(self, request):
        form = self.form_class(request.POST)

        if form.is_valid():
            user = form.save(commit=False)

            #cleaned (normalized) data    # Admin에 보이는 User 형식대로 저장하기 위해서 normalize 해줌
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']
            user.set_password(password)    # User password는 암호화되서 저장되기 때문에 이를 바꾸기 위해서는 이렇게 써줘야한다!
            user.save()
```

여기서 두가지의 request가 발생한다
1. Get - 작성양식을 url로 가져오는 것
2. Post - 입력한 내용을 보내는 것
이 두가지를 모두 실행하기 위해서 그에 맞는 로직이 필요한데
Class base view는 이것을 built-in function으로 정의해서 표현 할 수 있다는 것!

Python Django

2. User Authenticate & Login

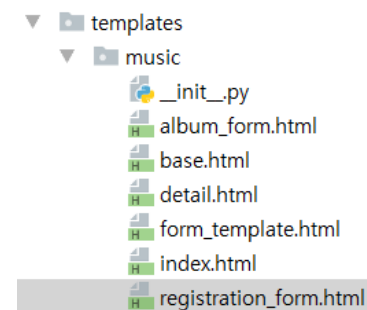
4) 로그인 기능 만들기 - views.py에 아래에 이어서 계속 적어주기

```
user = authenticate(username=username, password=password) # authenticate -> 해당 유저와 아이디가 존재하는지 확인하고 user로서 다시 return 해줌 없으면 none
if user is not None: #유저가 존재할경우
    if user.is_active:
        # 유저가 정지당하거나 로그인 불가능한 경우가 아닌 경우
        login(request, user) # login() 하나만으로 너무나도 간단히 로그인 기능 구현 할 수 있다는 것!!!
        return redirect('music:index')
```

5) Music의 urls.py로그인 url 생성하기

```
url(r'^register/$', views.UserFormView.as_view(), name='register'),
```

6) Templates / music 안에 이에 맞는 html 파일 생성하기 (registration_form)



Python Django

2. User Authenticate & Login

* Form의 구성이 add album 때와 같으므로 그냥 그대로 복사해주면 문제 없음!

7) album_form 내용을 복사해서 붙여넣고 수정 해주면 끝!

```
dels.py x detail.html x views.py x registration_form.html x forms.py x form_template.html x album_form.html x
div.container-fluid div.row div.col-sm-12.col-md-7 div.panel.panel-default
{% extends 'music/base.html' %}
{% block title %}Register{% endblock %}

{% block body %}<div class="container-fluid">

  <div class="row">
    <div class="col-sm-12 col-md-7">
      <div class="panel panel-default">
        <div class="panel-body">
          <h3>Create a New Account</h3>
          <form class="form-horizontal" action="" method="post" enctype="multipart/form-data">
            {% csrf_token %}
            {% include 'music/form_template.html' %}
            <div class="form-group">
              <div class="col-sm-offset-2 col-sm-10">
                <button type="submit" class="btn btn-success">Submit</button>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
{% endblock %}
```

완성!!!!!!

