

월말대비_0426

중요사항

```
from django.urls import path, include
from django import forms
from django.db import models
from django.conf import settings
```

media

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

메인 urls.py

```
from django.conf import settings
from django.conf.urls.static import static
```

```
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

이미지 저장하는 form.html

```
<forms enctype="multipart/form-data"></forms>
```

models.py

```
user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
```

인스타그램 만들기

modeling

1. Post 클래스 만들기

클래스 이름은 단수로 만들기

```
class Post(models.Model):
```

- image

```
image = models.ImageField(blank=True)
```

- content

```
from django.db import models
from django.conf import settings

# Create your models here.
class Post(models.Model):
    content = models.CharField(max_length=200)
    image = models.ImageField(blank=True)
```

posts 앱

CRUD

Create

- 이미지를 같이 받기 때문에 form을 받을때 `request.FILES`를 추가한다.

```
form = PostForm(request.POST, request.FILES)
```

- post에 옮겨서 저장(그냥 저장이 가능하지만 나중에 user를 따로 저장해야하기 때문에)

```
post = form.save(commit=False)
post.save()
```

- html에서 image를 넘길 때 enctype를 함께 넘겨야 한다.

```
<form enctype="multipart/form-data"></form>
```

- form에 저장할 때 `request.FILES` 요소가 필수적으로 들어간다.

```
form = PostForm(request.POST, request.FILES)
```

```
def create(request):
    if request.method=="POST":
        form = PostForm(request.POST, request.FILES)
        if form.is_valid():
            post = form.save(commit=False)
            post.save()
            return redirect('posts:index')
    else:
        form = PostForm()
        context = {
            'form': form,
        }
        return render(request, 'posts/form.html', context)
```

Read

- html에 이미지를 띄울 때 `post.image.url` 로 작성해야 한다.

```

```

```
index.html

{% extends 'base.html' %}

{% block body %}
<!--post를 보여주는 곳-->
<div class="row">
  <div class="col">
    {% for post in posts %}
      <div class="card">
        <div class="card-title">
          
        </div>
        <div class="card-body">
          {{ post.content }}
        </div>
      </div>
    {% endfor %}
  </div>
</div>

{% endblock %}
```

Update

- create와 마찬가지로 이미지를 보내기 위해 html파일의 form 태그에 enctype를 포함시켜야 한다.

```
<form enctype="multipart/form-data"></form>
```

- create와 마찬가지로 이미지를 저장하기 위해 views.py의 update함수에서 form에 `request.FILES` 요소를 추가해야 한다.

```
form = PostForm(request.POST, request.FILES)
```

```
def update(request, post_id):
    post = get_object_or_404(Post, pk=post_id)
    if request.method=="POST":
        form = PostForm(request.POST, request.FILES, instance=post)
        if form.is_valid():
            form.save()
            return redirect('posts:index')
    else:
        form = PostForm(instance=post)
        return render(request, 'posts/form.html', {'form':form})
```

Delete

- <a> 태그는 무조건 GET 방식이다.

```
def delete(request, post_id):
    post = get_object_or_404(Post, pk=post_id)
    post.delete()
    return redirect('posts:index')
```

accounts 앱

- forms.py

```
from django import forms
from django.conf import settings
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import get_user_model

class CustomUserCreationForm(UserCreationForm):
    class Meta:
        model = get_user_model()
```

댓글

CRUD

accounts