

5. url에 view 함수 연결해서 사진 출력하기

1. URL에 Photo View 연결

1. URL Resolver

- 이용자가 URL 요청 시 `urls.py` 에서 연결된 구현부를 실행
- URL과 구현부를 연결해주는 역할을 `views.py` 에서 수행
- `django/core/urllresolver.py` 모듈: URL Dispatch 역할
 1. `django/core/handlers/base.py` 의 `BaseHandler` 클래스가 URL로 요청(request) 받음
 2. `RegexURLResolver` 클래스로 URL 을 보냄
 3. `RegexURLResolver` 가 URL에 연결된 View를 찾아서 callback function, parameter 등의 덩어리를 `BaseHandler` 로 반환
 4. `BaseHandler` 에서 해당 함수 등의 덩어리를 실행하여 결과값 출력

2. 개별 사진 보기 View-1

- `photo/views.py`

```
from django.shortcuts import render
# render: 템플릿 출력물을 HttpResponse로 보내는 함수
from django.http import HttpResponse

def hello(request):
    return HttpResponse('안녕하세요!')
```

- `pystagram/urls.py` 에서 hello 함수 import, url pattern 추가

```
from django.conf.urls import url
from django.contrib import admin

from photos.views import hello # photos/views 모듈에서 hello import

urlpatterns = [
    url(r'^hello/$', hello),      # http://127.0.0.1:8000/hello
    url(r'^admin/', admin.site.urls),
]
```

3. urls.py

- `django/conf/urls` 모듈의 `url` 함수: URL 연결자를 만들어서 `urlpatterns` 에 넣는다

- `url` 함수 인자
 - **regex**: 주소 패턴(정규표현식) # `r'^hello/$`
 - **view**: 연결할 View (실행할 함수 객체) # `hello`
 - **name**: 주소 연결자 이름
 - **kwargs**: `urls` 에서 View로 전달할 dict형
- 개별 사진 보는 URL 생성

```
from django.conf.urls import url
from django.contrib import admin

from photos.views import hello
from photos.views import detail          # detail 추가

urlpatterns = [
    url(r'^hello/$', hello),
    url(r'^photos/(?P<pk>[0-9]+)/$',      # /photos/<사진 ID="">/
        detail, name='detail'),
    url(r'^admin/', admin.site.urls),
]
```

- `^photos/(?P<pk>[0-9]+)/$` 의미
 - `^`: 문자 뒤에 나열된 문자열로 시작
 - `[0-9]`: 0부터 9까지 범위에 속하는 문자
 - `+`: 앞에 지정한 문자열 패턴이 한번 이상 반복
 - `()`: 패턴 부분을 묶어냄(grouping)
 - `?P`: 묶어낸 패턴에 이름을 pk로 붙임
 - `$`: 문자 앞에 나열된 문자열로 끝
- `photos/views.py` 에 `detail` View 함수 추가

```
def detail(request):
    return HttpResponse('detail View 함수')
```

4. 개별 사진 보기 View-2

- `/photo/<사진 ID 숫자>` 에서 숫자가 **pk** 인자에 저장되어 `detail` View 함수로 전달됨

```
def detail(request, pk):
    msg = '{}번 사진 보여줄게요.'.format(pk)
    return HttpResponse(msg)
```

- `urls.py` url **photos/숫자** 형식으로 수정

```
url(r'^photos/([0-9]+)/$', detail, name='detail'),
```

2. Photo 모델에서 사진 정보를 가져와 출력하기

1. Photo 모델로 객체 찾기

```
from .models import Photo

def detail(request, pk):
    photo = Photo.objects.get(pk=pk)
    messages = (
        '<p>{pk}번 사진 보여줄게요</p>'.format(pk=photo.pk),
        '<p>주소는 {url}</p>'.format(url=photo.image.url),
    )
    # Photo 모델의 image 필드 url 속성으로 사진 URL 출력
    return HttpResponse('\n'.join(messages))
```

2. 찾는 객체가 없으면 404 오류 출력

- 해당 숫자(예:1023)이 없는 경우 **"DoesNotExist at photos/1023"** 예외처리

```
try:
    photo = Photo.objects.get(pk=pk)
except Photo.DoesNotExist:
    return HttpResponse("사진이 없습니다.")
```

- `views.py` 수정

from django.shortcuts import get_object_or_404

```
def detail(request, pk):
    photo = get_object_or_404(Photo, pk=pk)    # 추가
```

3. 업로드한 파일을 URL로 접근하기

- `img` 태그로 이미지 출력

```
messages = (
    '<p>{pk}번 사진 보여줄게요</p>'.format(pk=photo.pk),
    '<p>주소는 {url}</p>'.format(url=photo.image.url),
    '<p></p>'.format(url=photo.image.url),
)
```

- `MEDIA_URL`, `MEDIA_ROOT` 설정: `pystagram/setting.py`

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

- `models.py` 에 `upload_to` 옵션 변경

- ```
class Photo(models.Model):
 image = models.ImageField(upload_to='%Y/%m/%d/orig')
 filtered_image = models.ImageField(upload_to='%Y/%m/%d/filtered')
 content = models.TextField(max_length=500, null=True, blank=True)
 created_at = models.DateTimeField(auto_now_add=True)
```

- `url.py` 에 **static** 함수 추가

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

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

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