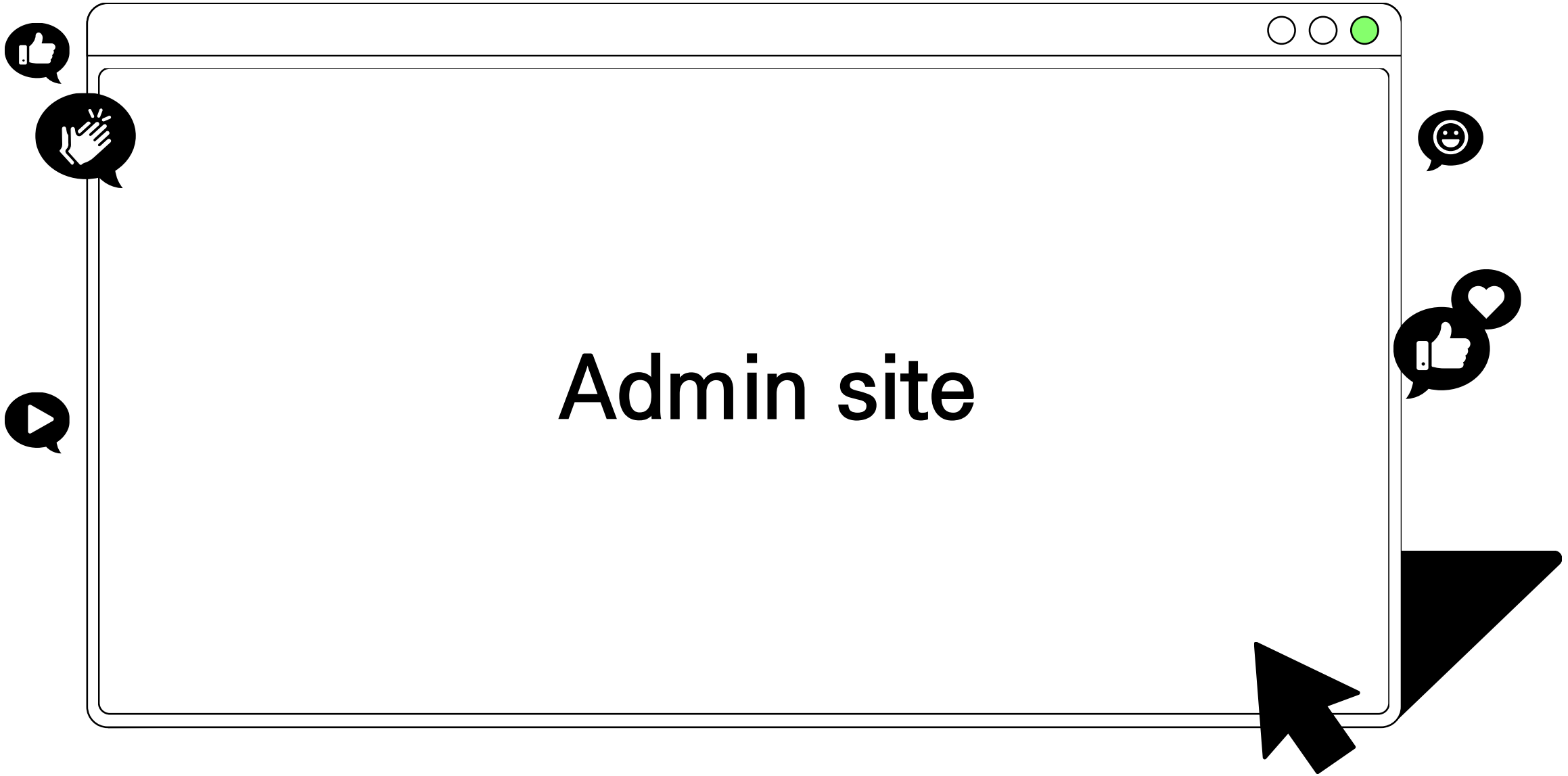
HAPPY HACKING



Admin site

HAPPY HACKING

개요

- Django의 가장 강력한 기능 중 하나인 automatic admin interface 알아보기
- “관리자 페이지”
  - 사용자가 아닌 서버의 관리자가 활용하기 위한 페이지
  - 모델 class를 admin.py에 등록하고 관리
  - 레코드 생성 여부 확인에 매우 유용하며 직접 레코드를 삽입할 수도 있음

HAPPY HACKING

admin 계정 생성

```
$ python manage.py createsuperuser
```

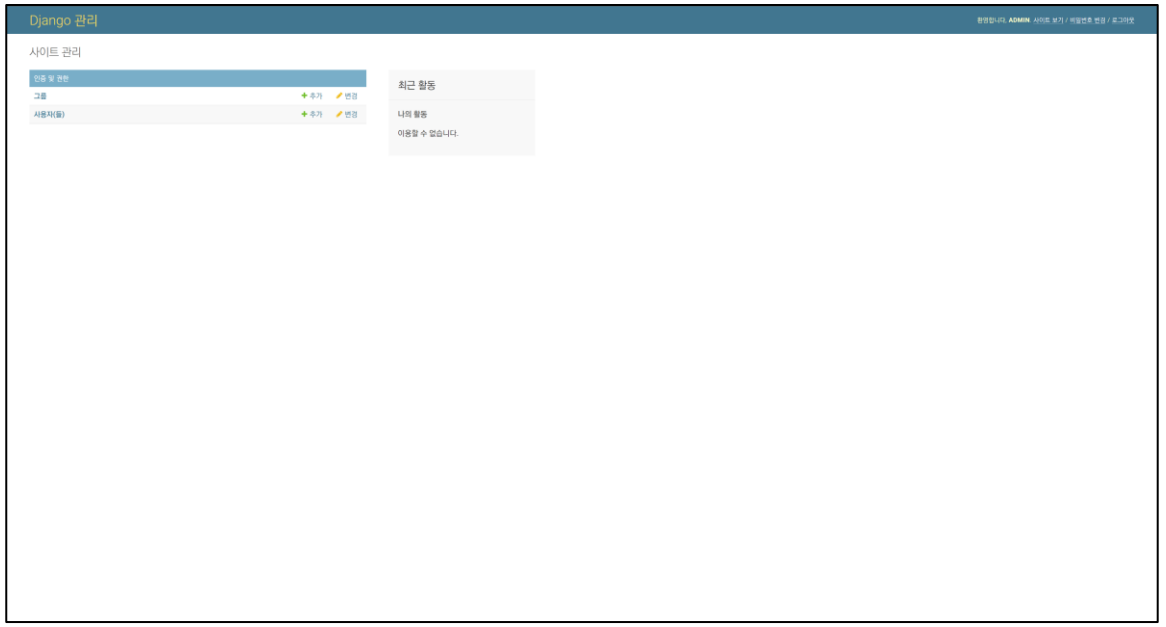
- username과 password를 입력해 관리자 계정을 생성

❖ email은 선택사항이기 때문에 입력하지 않고 enter를 입력하는 것이 가능
❖ 비밀번호 생성 시 보안상 터미널에 입력되지 않으니 무시하고 입력을 이어가도록 함

HAPPY HACKING

admin site 로그인

- http://127.0.0.1:8000/admin/ 로 접속 후 로그인
- 계정만 만든 경우 Django 관리자 화면에서 모델 클래스는 보이지 않음



HAPPY HACKING

admin에 모델 클래스 등록

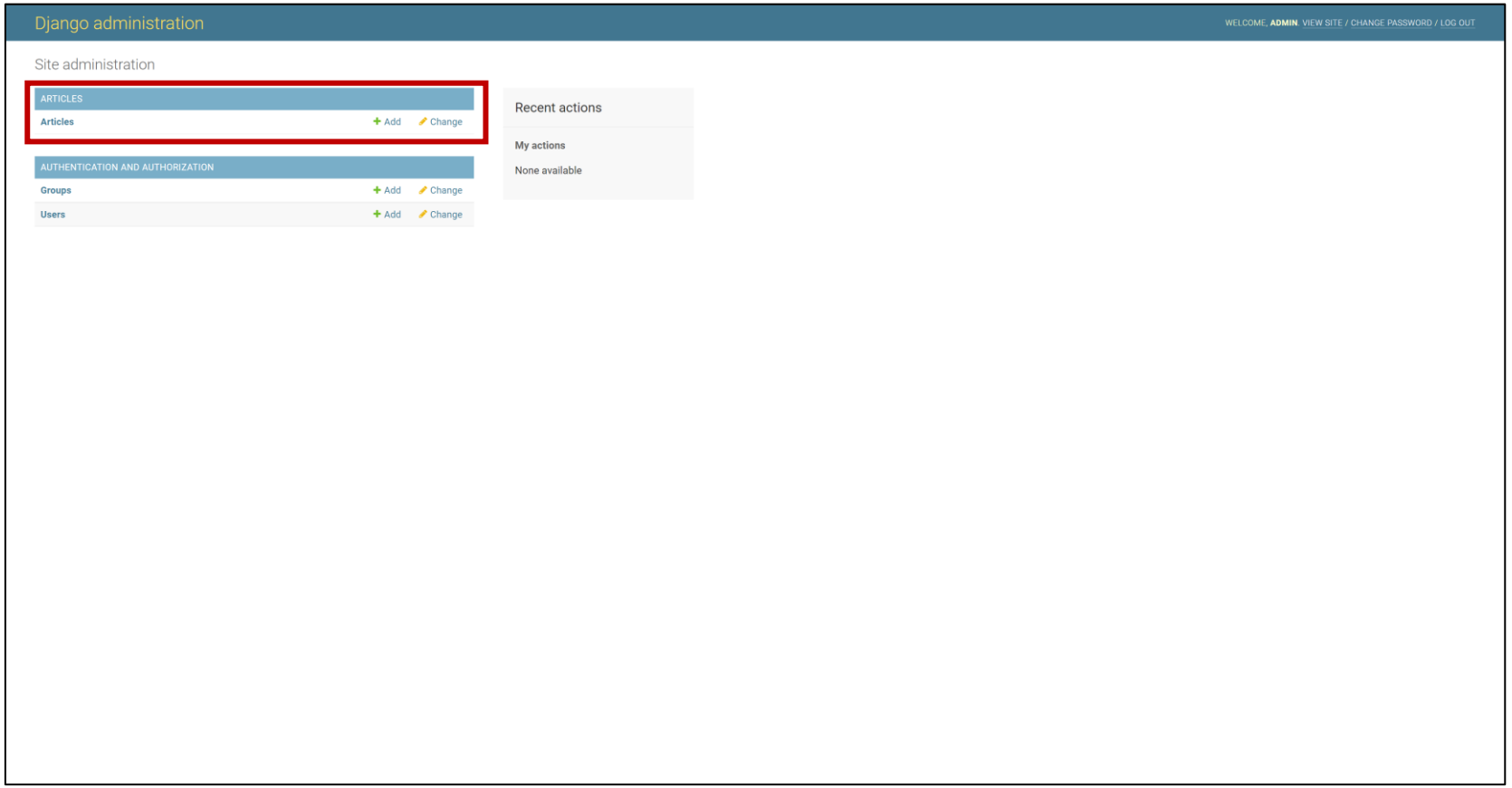
- 모델의 record를 보기 위해서는 admin.py에 등록 필요

```
# articles/admin.py

from django.contrib import admin
from .models import Article


admin.site.register(Article)
```

Admin site

HAPPY HACKING

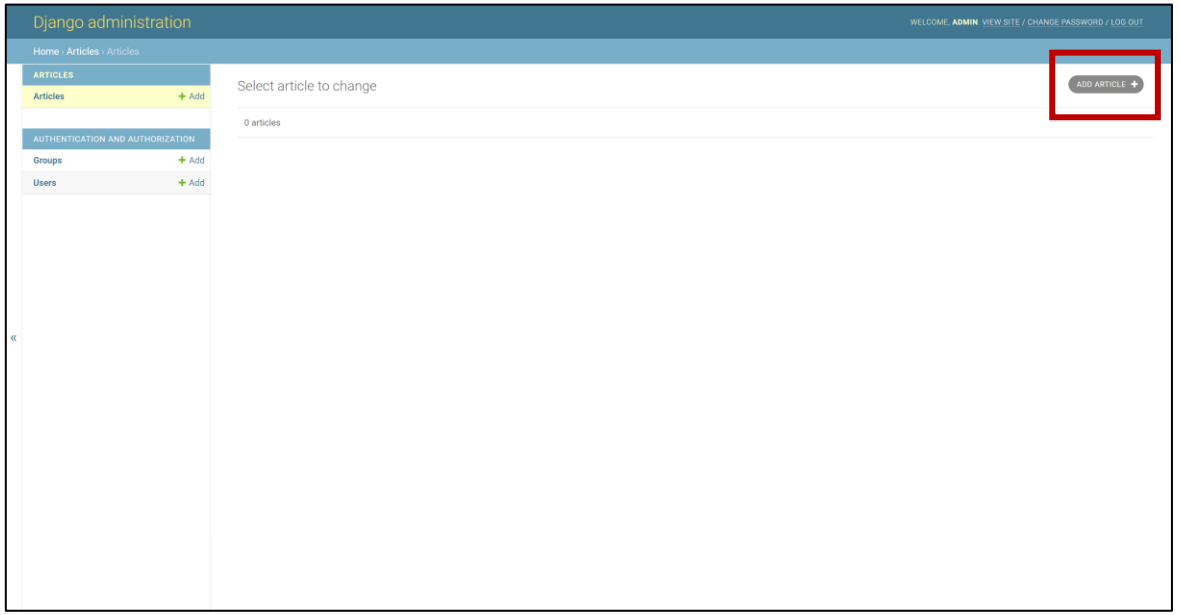
등록된 모델 클래스 확인

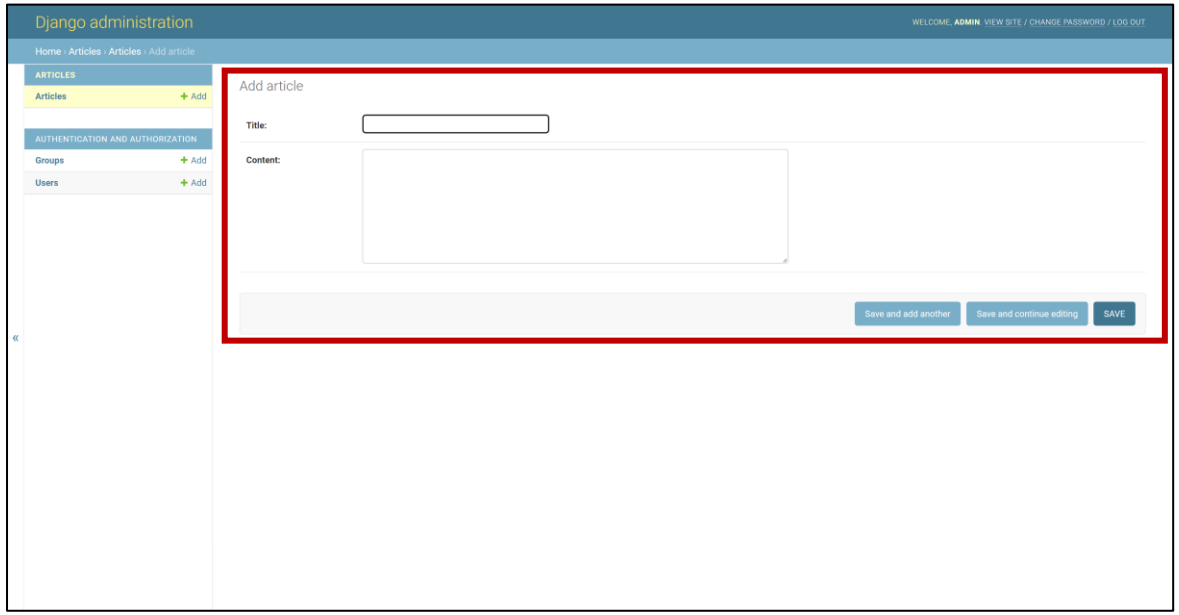


HAPPY HACKING

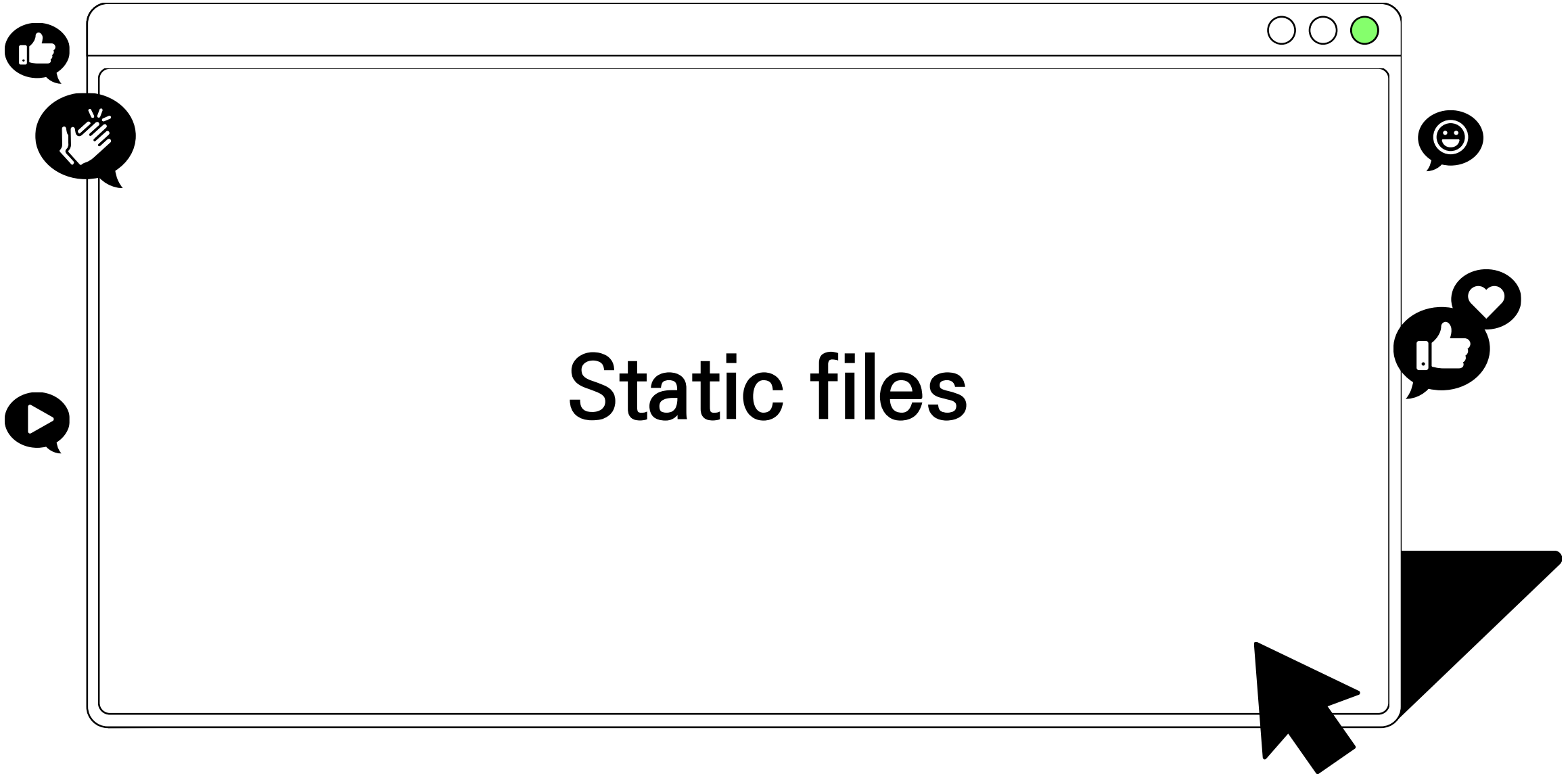
데이터 CRUD 테스트

- admin 페이지에서 데이터를 조작해보기





HAPPY HACKING



Static files

HAPPY HACKING

웹서버와 정적 파일

- 웹 서버는 특정 위치(URL)에 있는 자원(resource)을 요청(HTTP request) 받아서 제공(serving)하는 응답(HTTP response)을 처리하는 것을 기본 동작으로 함
- 즉, 웹 서버는 요청 받은 URL로 서버에 존재하는 정적 자원(static resource)를 제공

HAPPY HACKING

정적 파일

- 응답할 때 별도의 처리 없이 파일 내용을 그대로 보여주면 되는 파일
  - 사용자의 요청에 따라 내용이 바뀌는 것이 아니라 요청한 것을 그대로 보여주는 파일
- 예를 들어, 웹 서버는 일반적으로 이미지, 자바 스크립트 또는 CSS와 같은 미리 준비된 추가 파일(움직이지 않는)을 제공해야 함
- 파일 자체가 고정되어 있고, 서비스 중에도 추가되거나 변경되지 않고 고정되어 있음
- Django에서는 이러한 파일들을 “Static file”이라 함
  - Django는 staticfiles 앱을 통해 정적 파일과 관련 된 기능을 제공

https://docs.djangoproject.com/en/3.2/howto/static-files/#managing-static-files-e-g-images-javascript-css

HAPPY HACKING

정적 파일 활용

- django.contrib.staticfiles가 INSTALLED_APPS에 포함되어 있는지 확인
- settings.py에서 STATIC_URL을 정의
- 템플릿에서 static 템플릿 태그를 사용하여 지정된 상대경로에 대한 URL을 빌드

```
{% load static %}

<img src="{% static 'my_app/example.jpg' %}" alt="My image">
```

- 앱의 static 디렉토리에 정적 파일을 저장
  - 예시) my_app/static/my_app/example.jpg

https://docs.djangoproject.com/en/3.2/howto/static-files/#configuring-static-files

Static files

HAPPY HACKING

- STATICFILES_DIRS
  - ‘app/static/’ 디렉토리 경로(기본 경로) 를 사용하는 것외에
    추가적인 정적 파일 경로 목록을 정의하는 리스트
  - 추가 파일 디렉토리에 대한 전체 경로를 포함하는 문자열 목록으로 작성되어야 함

```
STATICFILES_DIRS = [
    BASE_DIR / 'static',
]
```

https://docs.djangoproject.com/en/3.2/ref/settings/#staticfiles-dirs

Static files

HAPPY HACKING

- STATIC_URL
  - STATIC_ROOT에 있는 정적 파일을 참조 할 때 사용할 URL
    - 개발 단계에서는 실제 정적 파일들이 저장되어 있는 'app/static/' 경로(기본 경로) 및 STATICFILES_DIRS에 정의된 추가 경로들을 탐색함
  - 실제 파일이나 디렉토리가 아니며, URL로만 존재
  - 비어 있지 않은 값으로 설정 한다면 반드시 slash(/)로 끝나야 함

```
STATIC_URL = '/static/'
```

https://docs.djangoproject.com/en/3.2/ref/settings/#static-url

Static files

HAPPY HACKING

- STATIC_ROOT
  - collectstatic이 배포를 위해 정적 파일을 수집하는 디렉토리의 절대 경로
  - django 프로젝트에서 사용하는 모든 정적 파일을 한 곳에 모아 넣는 경로
  - 개발 과정에서 settings.py의 DEBUG 값이 True로 설정되어 있으면 해당 값은 작용되지 않음
    - 직접 작성하지 않으면 django 프로젝트에서는 settings.py에 작성되어 있지 않음
  - 실 서비스 환경(배포 환경)에서 django의 모든 정적 파일을 다른 웹 서버가 직접 제공하기 위함

https://docs.djangoproject.com/en/3.2/ref/settings/#static-root
https://docs.djangoproject.com/en/3.2/ref/contrib/staticfiles/#collectstatic

Static files

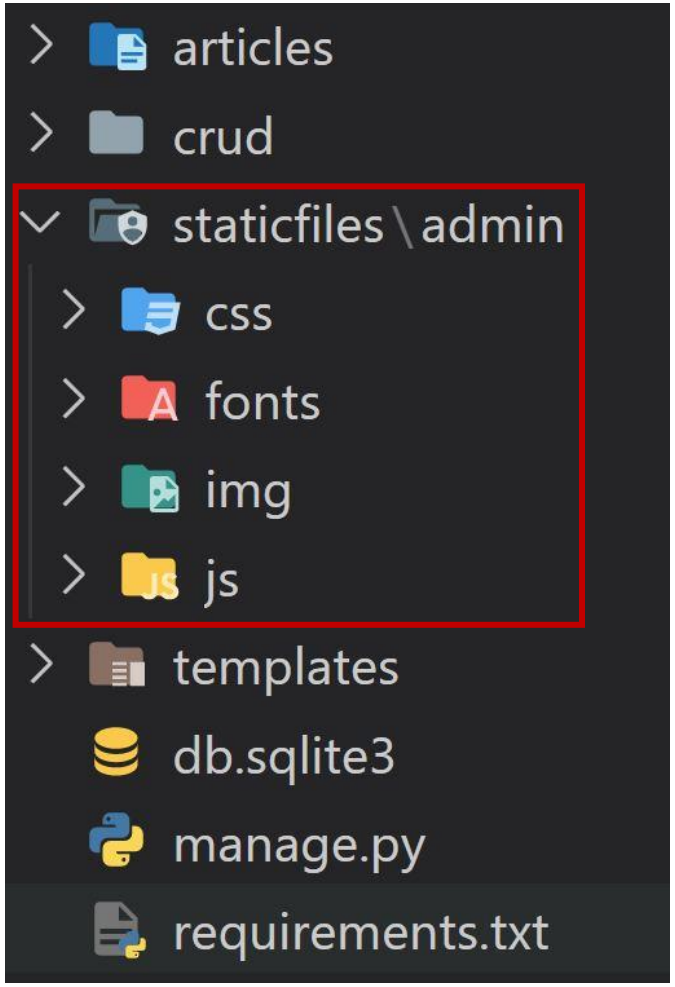
- [참고] collectstatic

```
STATIC_ROOT = BASE_DIR / 'staticfiles'
```

STATIC_ROOT 작성

```
$ python manage.py collectstatic
```

collectstatic 명령어



실행 결과

*개발 환경에서는 사용하지 않으니 삭제 후 진행

https://docs.djangoproject.com/en/3.2/ref/settings/#static-root

https://docs.djangoproject.com/en/3.2/ref/contrib/staticfiles/#collectstatic

Static files

HAPPY HACKING

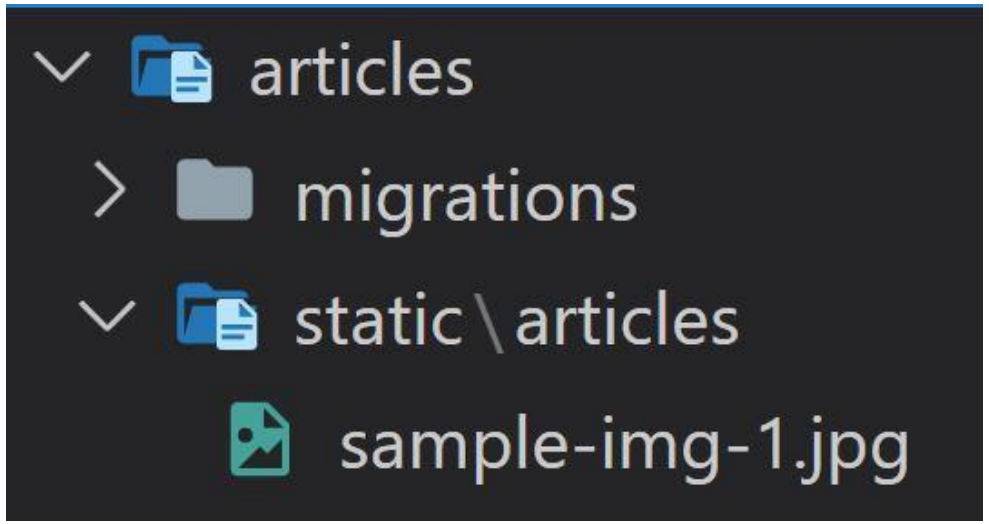
- load
  - 사용자 정의 템플릿 태그 세트를 로드(load)
  - 로드하는 라이브러리, 패키지에 등록된 모든 태그와 필터를 불러옴
- static
  - STATIC_ROOT에 저장된 정적 파일에 연결

```
{% load static %}

<img src="{% static 'my_app/example.jpg' %}" alt="My image">
```

https://docs.djangoproject.com/en/3.2/ref/templates/builtins/#load
https://docs.djangoproject.com/en/3.2/ref/templates/builtins/#static

HAPPY HACKING

정적 파일 사용하기 (1/3) – 기본 경로



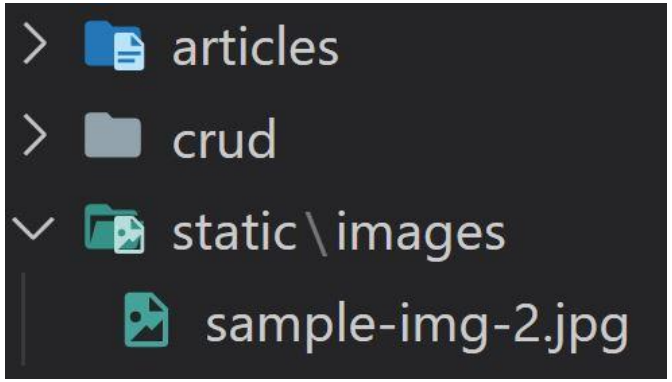
정적 파일 경로
app/static/app/

```
<!-- articles/detail.html -->

{% extends 'base.html' %}
{% load static %}

{% block content %}
  <img src="{% static 'articles/sample-img-1.jpg' %}" alt="sample-img">
  <h2>DETAIL</h2>
  ...
{% endblock content %}
```

template에서 경로 참조

HAPPY HACKING

정적 파일 사용하기 (2/3) – 추가 경로




```python
STATICFILES_DIRS = [
    BASE_DIR / 'static',
]
```

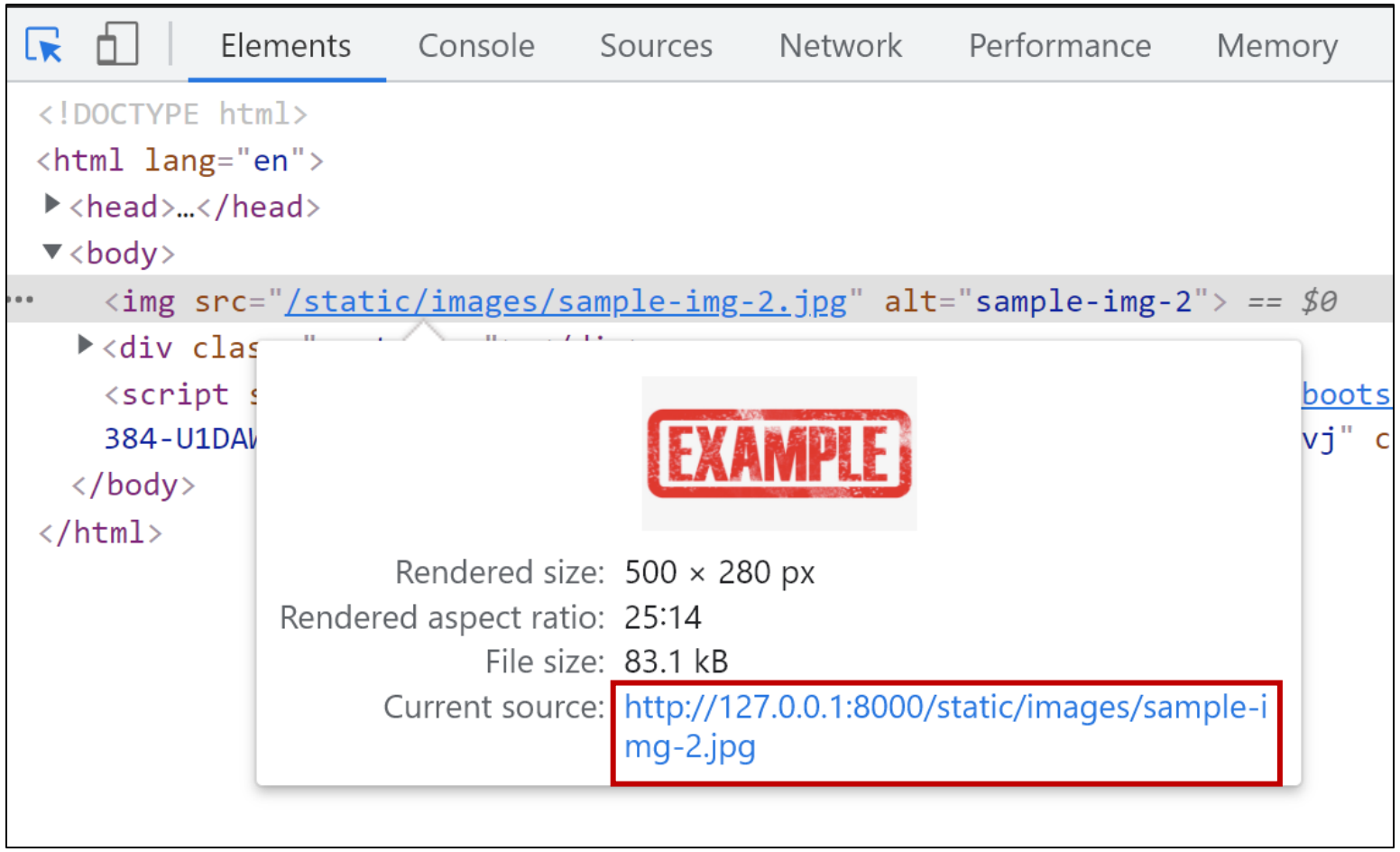
정적 파일 위치 및
추가 경로 작성

```html
<!-- base.html -->

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
  ...
</head>
<body>
  <img src="{% static 'images/sample-img-2.jpg' %}" alt="sample-img-2">
  <div class="container">
    {% block content %}
    {% endblock content %}
  </div>
</body>
</html>
```

template에서 경로 참조

HAPPY HACKING

정적 파일 사용하기 (3/3) – STATIC_URL 확인



크롬 개발자 도구 확인