

Django 웹 프레임워크

일반적인 특징

일반적인 특징

◦ MVC 패턴 기반 MVT

- 장고는 MVC Model-View-Controller를 기반으로 한 프레임워크
- 장고에서는 View를 Template, Controller를 View라고 함

◦ 객체 관계 매핑

- 장고의 객체 관계 매핑 ORM, Object-Relational Mapping은 데이터베이스 시스템과 데이터 모델 클래스를 연결시키는 다리과 같은 역할

일반적인 특징

일반적인 특징

- 자동으로 구성되는 관리자 화면

- 장고는 웹 서버의 콘텐츠, 즉 데이터베이스에 대한 관리 기능을 위하여 프로젝트를 시작하는 시점에 기본 기능으로 관리자 화면을 제공

- 우아한 URL 설계

- 웹 프로그래밍에서 URL 디자인은 필수인데, 장고에서는 유연하면서도 강력한 기능을 제공. 장고에서는 우아한 Elegant URL 방식을 채택하여 URL을 직관적이고 쉽게 표현

일반적인 특징

일반적인 특징

◦ 자체 템플릿 시스템

- 장고는 내부적으로 확장이 가능하고 디자인이 쉬운 강력한 템플릿 시스템을 갖고 있음.

이를 통해 화면 디자인과 로직에 대한 코딩을 분리하여 독립적으로 개발 진행

◦ 캐시 시스템

- 동적인 페이지를 만들기 위해서 데이터베이스 쿼리를 수행하고 템플릿을 해석하며, 관련 로직을 실행해서 페이지를 생성하는 일은 서버에 엄청난 부하를 주는 작업

일반적인 특징

일반적인 특징

◦ 다국어 지원

- 다국어 장고는 동일한 소스코드를 다른 나라에서도 사용할 수 있도록 텍스트의 번역, 날짜/시간/숫자의 포맷, 타임존의 지정 등과 같은 다국어 환경을 제공

◦ 소스 변경사항 자동 반영

- 장고에서는 *.py 파일의 변경 여부를 감시하고 있다가 변경이 되면 실행 파일에 변경 내역을 바로 반영

장고 프로그램 설치

윈도우에서 장고 설치

- 파이썬 3.x 버전을 설치하면 pip 프로그램도 같이 설치.
- pip(Python Install Package) 프로그램은 파이썬의 오픈소스 저장소인 PyPI(Python Package Index)에 있는 SW 패키지를 설치하고 관리해주는 명령
- (base) C:\MyTest> **pip install Django**

```
(base) C:\MyTest>pip install Django
Collecting Django
  Downloading Django-3.2.6-py3-none-any.whl (7.9 MB)
    | 7.9 MB 1.6 MB/s
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.4.1-py3-none-any.whl (25 kB)
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.1-py3-none-any.whl (42 kB)
    | 42 kB 393 kB/s
Requirement already satisfied: pytz in c:\programdata\anaconda3\lib\site-packages (from Django) (2020.1)
Installing collected packages: asgiref, sqlparse, Django
Successfully installed Django-3.2.6 asgiref-3.4.1 sqlparse-0.4.1
```

장고 프로그램 설치

장고 프로그램 설치 확인

- 장고는 파이썬 환경에서 동작하는 패키지이므로, 장고가 정상적으로 설치되었는지 확인하기 위해서 아래와 같이 명령을 입력
- (base) C:\WMyTest>python -m django --version

```
(base) C:\WMyTest>python -m django --version
3.2.6
```

```
(base) C:\WMyTest>python -m django --version
C:\ProgramData\Anaconda3\python.exe: No module named django
```

```
(base) C:\WMyTest>python -c "import django; print(django.__path__)"
['C:\ProgramData\Anaconda3\lib\site-packages\django']
```

장고에서의 애플리케이션 개발 방식

장고에서의 애플리케이션 개발 방식

- 웹 사이트의 전체 프로그램 또는 모듈화된 단위 프로그램을 애플리케이션
- 이션
- 즉, 프로그램으로 코딩할 대상을 애플리케이션이라고 부름
- 사이트에 대한 전체 프로그램을 프로젝트(Project)라고 함
- 모듈화된 단위 프로그램을 애플리케이션(Application)이라 부름

장고에서의 애플리케이션 개발 방식

MVC 패턴 기반 MVT

- 일반적인 MVC 패턴
- 개발 시 일반적으로 언급되는 MVC(Model-View-Controller) 패턴이란 데이터 Model, 사용자 인터페이스View, 데이터를 처리하는 로직 Controller을 구분해서 한 요소가 다른 요소들에 영향을 주지 않도록 설계하는 방식

장고에서의 애플리케이션 개발 방식

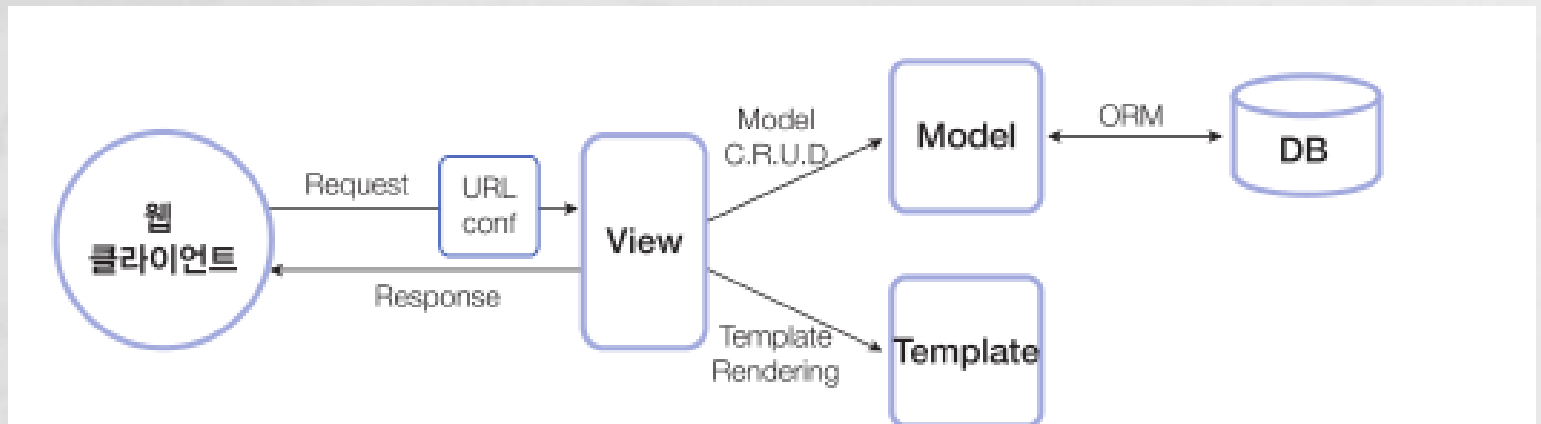
MVC 패턴 기반 MVT

- 장고의 MVT 패턴
- View를 Template, Controller는 View라고 표현
- MVC 대신에 MVT(Model-View-Template)패턴이라고 함
- 모델(Model)은 데이터베이스에 저장되는 데이터를 의미
- 템플릿(Template)은 사용자에게 보여지는 UI부분
- 뷰(View)는 실질적으로 프로그램 로직이 동작하여 데이터를 가져오고 적절하게 처리한 결과를 템플릿에 전달하는 역할을 수행

장고에서의 애플리케이션 개발 방식

MVC 패턴 기반 MVT

- 장고의 MVT 패턴



장고에서의 애플리케이션 개발 방식

장고 MVT 패턴

◦ 장고의 MVT 패턴

- 클라이언트로부터 요청을 받으면 URL.conf를 이용하여 URL을 분석
- URL 분석 결과를 통해 해당 URL에 대한 처리를 담당할 **뷰**를 결정
- 뷰는 자신의 로직을 실행하면서, 만일 데이터베이스 처리가 필요하면 **모델**을 통해 처리하고 그 결과를 반환
- 뷰는 자신의 로직 처리가 끝나면 **템플릿**을 사용하여 클라이언트에 전송할 HTML 파일을 생성
- 뷰는 최종 결과로 HTML 파일을 클라이언트에게 보내 응답

장고에서의 애플리케이션 개발 방식

Model – 데이터베이스 정의(models.py)

- 모델이란 사용될 데이터에 대한 정의를 담고 있는 장고의 클래스
- 장고는 ORM 기법을 사용하여 애플리케이션에서 사용할 데이터베이스를 클래스로 매핑해서 코딩할 수 있음
- 장고는 테이블 및 컬럼을 자동으로 생성하기 위해 필요한 많은 규칙

```
from django.db import models

class Person(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```

```
CREATE TABLE myapp_person (
    "id" serial NOT NULL PRIMARY KEY,
    "first_name" varchar(30) NOT NULL,
    "last_name" varchar(30) NOT NULL
);
```

장고에서의 애플리케이션 개발 방식

URL.conf – URL 정의 (urls.py)

- 파이썬의 URL 정의 방식은 전통적인 자바나 PHP 계열의 URL보다 직관적이고 이해하기가 쉬움.
- 이런 방식을 우아한(Elegant URL)이라고 부르는 것.
- URL을 정의하기 위해서는 urls.py 파일에 URL과 처리 함수(뷰 View라고 부름)를 매핑하는 파이썬 코드를 작성하면 됨

```
from django.urls import path

from . import views

urlpatterns = [
    path('articles/2003/', views.special_case_2003),
    path('articles/<int:year>/', views.year_archive),
    path('articles/<int:year><int:month>/', views.month_archive),
    path('articles/<int:year><int:month>/<slug:slug>/', views.article_detail),
]
```

장고에서의 애플리케이션 개발 방식

웹 클라이언트가 웹 서버에 페이지 요청 시

- 웹 클라이언트가 웹 서버에 페이지 요청 시
- 장고에서 URL을 분석하는 순서
 - Setting.py파일의 ROOT_URLCONF 항목을 읽어 최상위 URLconf(urls.py)의 위치 파악
 - URLconf를 로딩하여 urlpatterns 변수에 지정되어 있는 URL리스트를 검사
 - 위에서부터 순서대로 URL 리스트의 내용을 검사하면서 URL 패턴이 매치되면 검사를 종료
 - 매치된 URL의 뷰를 호출 - 여기서 뷰는 함수 또는 클래스의 메소드
 - 호출 시 HttpRequest 객체와 매칭할 때 추출된 단어들을 뷰에 인자로 넘겨줌
 - URL 리스트 끝까지 검사했는데도 매칭에 실패하면 에러를 처리하는 뷰를 호출

장고에서의 애플리케이션 개발 방식

URL 패턴을 정의

- URL 패턴의 일부 문자열을 추출하기 위한 것이며
- `<type:name>` 형식으로 사용
- 장고에서는 Path Converter
 - `str` : /(슬래시)를 제외한 모든 문자열과 매치. 타입이 지정되지 않았다면 디폴트로 `str` 타입을 사용
 - `Int` : 0 또는 양의 정수와 매치 매치된 정수는 파이썬의 `int` 타입으로 변환
 - `slug` : `slug` 형식의 문자열(ASCII, 숫자, 하이픈, 밑줄로만 구성됨)과 매치
 - `uuid` : `UUID` 형식의 문자열과 매치. 매치된 문자열은 파이썬의 `UUID` 타입으로 변환
 - `path` :/(슬래시)를 포함한 모든 문자열과 매치 이는 URL 패턴의 일부가 아니라 전체를 추출하고자 할 때 많이 사용

장고에서의 애플리케이션 개발 방식

URL 패턴을 정의

◦ 정규표현식에 사용되는 문자들

표현	의미
.	(Dot) 모든 문자 하나(any single character)
^	(Caret) 문자열의 시작
\$	문자열의 끝
[]	[] 괄호에 있는 문자 하나, 예를 들어 [akz]라면 a 또는 k 또는 z
[^]	[] 괄호에 있는 문자 이외의 문자 하나, 만일 [^ab]라면 a와 b를 제외한 문자 하나
*	0번 이상 반복, {0,}와 동일
+	1번 이상 반복, {1,}와 동일
?	0번 또는 1번 반복, {0,1}과 동일
{n}	n번 반복
{m,n}	최소 m번에서 최대 n번까지 반복
	예를 들어 A B 라면, A 또는 B
[a-z]	a에서 z까지 임의의 문자, 즉 영문 소문자 한 개
[w]	영문, 숫자 또는 밑줄(_) 한 개, [0-9a-zA-Z_]와 동일
[d]	숫자 한 개, [0-9]와 동일

장고에서의 애플리케이션 개발 방식

View – 로직 정의 (view.py)

- 뷰는 웹 요청을 받아서 데이터베이스 접속 등 해당 애플리케이션의 로직에 맞는 처리를 하고, 그 결과 데이터를 HTML로 변환하기 위하여 템플릿 처리를 한 후에, 최종 HTML로 된 응답 데이터를 웹 클라이언트로 반환하는 역할
- 간단한 뷰 - 현재의 날짜와 시간 변환

```
from django.http import HttpResponse
import datetime

def current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body>It is now %s.</body></html>" % now
    return HttpResponse(html)
```

장고에서의 애플리케이션 개발 방식

Template – 화면 UI 정의 (*.html)

- 개발자가 응답에 사용할 *.html 파일을 작성하면, 장고는 이를 해석해서 최종 HTML 텍스트 응답을 생성하고, 이를 클라이언트에게 보내줌
- 클라이언트(보통 웹 브라우저)는 응답으로 받은 HTML 텍스트를 해석해서 우리가 보는 웹 브라우저 화면에 UI를 보여주는 것
- 템플릿 파일은 *.html 확장자를 가지며, 장고의 템플릿 시스템 문법에 맞게 작성
- 장고에서 템플릿 파일을 찾을 때는 TEMPLATES 및 INSTALLED_APPS에서 지정된 앱의 디렉토리를 검색.
- 이 항목들은 프로젝트 설정 파일인 settings.py 파일에 정의되어 있음

장고에서의 애플리케이션 개발 방식

MVT 코딩 순서

- 모델, 뷰, 템플릿 셋 중에서 무엇을 먼저 코딩해야 하는지에 대해 정해진 순서는 없음.
- MVT 방식에 따르면 화면 설계는 뷰와 템플릿 코딩으로 연결되고, 테이블 설계는 모델 코딩에 반영.
- 그렇기 때문에 독립적으로 개발할 수 있는 모델을 먼저 코딩하고, 뷰와 템플릿은 서로 영향을 미치므로 모델 이후에 같이 코딩하는 것이 일반적

장고에서의 애플리케이션 개발 방식

MVT 코딩 순서

- **프로젝트 뼈대 만들기** : 프로젝트 및 앱 개발에 필요한 디렉토리와 파일 생성
- **모델 코딩하기** : 테이블 관련 사항을 개발(models.py, admin.py 파일)
- **URLconf 코딩하기** : URL 및 뷰 매핑 관계를 정의(urls.py 파일)
- **템플릿 코딩하기** : 화면 UI 개발(templates/ 디렉토리 하위의 *.html 파일들)
- **뷰 코딩하기** : 애플리케이션 로직 개발(views.py 파일)

애플리케이션 설계

애플리케이션 설계

- index.html : 최근에 실시하고 있는 질문의 리스트를 보여줌.
- detail.html : 하나의 질문에 대해 투표할 수 있도록 답변 항목을 폼으로 보여줌
- results.html : 질문에 따른 투표 결과를 보여줌



애플리케이션 설계

애플리케이션 설계

- index.html : 최근에 실시하고 있는 질문의 리스트를 보여줌.
- detail.html : 하나의 질문에 대해 투표할 수 있도록 답변 항목을 폼으로 보여줌
- results.html : 질문에 따른 투표 결과를 보여줌



애플리케이션 설계

애플리케이션 설계

◦ Question 테이블 설계

컬럼명	타입	제약 조건	설명
id	integer	NotNull, PK, AutoIncrement	Primary Key
question_text	varchar(200)	NotNull	질문 문장
pub_date	datetime	NotNull	질문 생성 시각

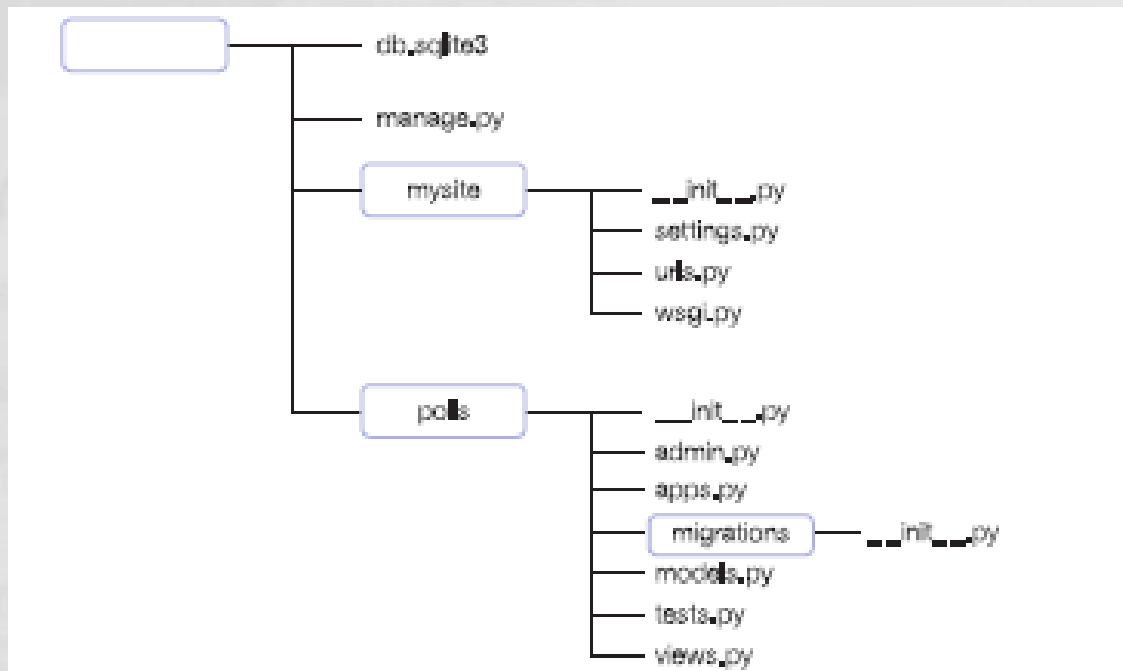
◦ Choice 테이블 설계

컬럼명	타입	제약 조건	설명
id	integer	NotNull, PK, AutoIncrement	Primary Key
choice_text	varchar(200)	NotNull	답변 항목 문구
votes	integer	NotNull	투표 카운트
question	integer	NotNull, FK (Questionid), Index	Foreign Key

프로젝트 뼈대 만들기

프로젝트 뼈대 만들기

- 프로젝트 뼈대의 최종 디렉토리



프로젝트 뼈대 만들기

프로젝트 뼈대 만들기

프로젝트 뼈대의 최종 디렉토리

디렉토리	프로젝트 관련 디렉토리 및 파일을 모아주는 최상위 루트 디렉토리입니다. 보통 settings.py 파일의 BASE_DIR 항목으로 지정됩니다.
db.sqlite3	SQLite3 데이터베이스 파일입니다. 테이블이 들어있습니다.
manage.py	장고의 명령어를 처리하는 파일입니다.
mysite 디렉토리	프로젝트명으로 만들어진 디렉토리입니다. 프로젝트 관련 파일들이 들어있습니다.
__init__.py	디렉토리에 이 파일이 있으면 파이썬 패키지로 인식합니다.
settings.py	프로젝트 설정 파일입니다.
urls.py	프로젝트 레벨의 URL 패턴을 정의하는 최상위 URLconf입니다. 보통은 애플리케이션 디렉토리마다 하위 urls.py 파일이 또 있습니다.
wsgi.py	Apache와 같은 웹 서버와 WSGI 규격으로 연동하기 위한 파일입니다.

프로젝트 뼈대 만들기

프로젝트 뼈대 만들기

프로젝트 뼈대의 최종 디렉토리

polls 디렉토리	애플리케이션명으로 만들어진 애플리케이션 디렉토리입니다. 해당 애플리케이션 관련 파일들이 들어있습니다.
__init__.py	디렉토리에 이 파일이 있으면 파이썬 패키지로 인식합니다.
admin.py	Admin 사이트에 모델 클래스를 등록해주는 파일입니다.
apps.py	애플리케이션의 설정 클래스를 정의하는 파일입니다.
migrations 디렉토리	데이터베이스 변경사항을 관리하기 위한 디렉토리입니다. 데이터베이스에 추가 삭제 변경 등이 발생하면 변경 내역을 기록한 파일들이 위치합니다.
models.py	데이터베이스 모델 클래스를 정의하는 파일입니다.
tests.py	단위 테스트용 파일입니다. 이 책에서는 사용하지 않습니다.
views.py	뷰 함수를 정의하는 파일입니다. 함수형 뷰 및 클래스형 뷰 모두 이 파일에 정의합니다.

프로젝트 뼈대 만들기

프로젝트 뼈대 만들기

프로젝트 뼈대의 최종 디렉토리

templates 디렉토리	프로젝트를 진행하면서 추가됩니다. 템플릿 파일들이 들어 있습니다. 보통은 프로젝트 레벨과 애플리케이션 레벨의 템플릿으로 구분하여 ch3/templates 및 ch3/polls/templates 위치에 생성됩니다.
static 디렉토리	프로젝트를 진행하면서 추가됩니다. CSS, Image, Javascript 파일들이 들어 있습니다. 보통은 프로젝트 레벨과 애플리케이션 레벨로 구분하여 ch3/static 및 ch3/polls/static 위치에 생성됩니다.
logs 디렉토리	프로젝트를 진행하면서 추가됩니다. 로그 파일들이 들어 있습니다. 로그 파일의 위치는 settings.py 파일의 LOGGING 항목으로 지정합니다.

프로젝트 뼈대 만들기

프로젝트 뼈대 만들기 순서 명령

- > `django-admin startproject mysite` // mysite라는 프로젝트를 생성함
- > `python manage.py startapp polls` // polls라는 애플리케이션을 생성
- > `notepad settings.py` //설정 파일을 확인 및 수정
- > `python manage.py migrate` //데이터베이스에 기본 테이블을 생성
- > `python manage.py runserver` //현재까지 작업을 개발용 웹 서버로

확인

프로젝트 뼈대 만들기

프로젝트 생성

- mysite라는 프로젝트 만듦
- (base) C:\MyTest>django-admin startproject mysite

```
(base) C:\MyTest>django-admin startproject mysite

(base) C:\MyTest>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest 디렉터리

2021-08-23 오후 08:02 <DIR> .
2021-08-23 오후 08:02 <DIR> ..
2021-08-23 오후 06:44 <DIR> cgi-server
2021-08-23 오후 06:42 369 cgi_client.py
2021-08-23 오후 08:02 <DIR> mysite
2020-09-07 오전 08:19 497 my_httpserver.py
2021-08-23 오후 06:49 <DIR> wsgi-server
                2개 파일            866 바이트
                5개 디렉터리 21,002,899,456 바이트 남음
```

프로젝트 뼈대 만들기

프로젝트 생성

o mysite 구성

```
(base) C:\MyTest>cd mysite

(base) C:\MyTest\mysite>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest\mysite 디렉터리

2021-08-23 오후 08:02 <DIR>          .
2021-08-23 오후 08:02 <DIR>          ..
2021-08-23 오후 08:02                684 manage.py
2021-08-23 오후 08:02 <DIR>          mysite
                        1개 파일              684 바이트
                        3개 디렉터리  21,001,031,680 바이트 남음

(base) C:\MyTest\mysite>cd mysite

(base) C:\MyTest\mysite\mysite>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest\mysite\mysite 디렉터리

2021-08-23 오후 08:02 <DIR>          .
2021-08-23 오후 08:02 <DIR>          ..
2021-08-23 오후 08:02                405 asgi.py
2021-08-23 오후 08:02            3,364 settings.py
2021-08-23 오후 08:02                769 urls.py
2021-08-23 오후 08:02                405 wsgi.py
2021-08-23 오후 08:02                 0 __init__.py
                        5개 파일              4,943 바이트
                        2개 디렉터리  21,000,966,144 바이트 남음
```


프로젝트 뼈대 만들기

프로젝트 생성

- mysite 디렉토리 이름 변경
- (base) C:\MyTest>move mysite projectsite

```
(base) C:\MyTest\mysite>cd ..  
(base) C:\MyTest>cd ..  
(base) C:\MyTest>move mysite projectsite  
1개의 디렉토리를 이동했습니다.  
(base) C:\MyTest>
```

```
(base) C:\MyTest>dir  
C 드라이브의 볼륨에는 이름이 없습니다.  
볼륨 일련 번호: 421A-9EBA  
  
C:\MyTest 디렉터리  
  
2021-08-23 오후 08:08 <DIR> .  
2021-08-23 오후 08:08 <DIR> ..  
2021-08-23 오후 06:44 <DIR> cgi-server  
2021-08-23 오후 06:42 369 cgi_client.py  
2020-09-07 오전 08:19 497 my_httpserver.py  
2021-08-23 오후 08:02 <DIR> projectsite  
2021-08-23 오후 06:49 <DIR> wsgi-server  
2개 파일 866 바이트  
5개 디렉터리 21,002,772,480 바이트 남음
```


프로젝트 뼈대 만들기

애플리케이션 생성

- 프로젝트 루트 디렉토리 projectsite으로 이동해서 polls라는 애플리케이션을 만드는 명령을 실행
- (base) C:\MyTest>cd projectsite
- (base) C:\MyTest\projectsite>python manage.py startapp polls

```
(base) C:\MyTest>cd projectsite
(base) C:\MyTest\projectsite>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest\projectsite 디렉터리
2021-08-23 오후 08:02 <DIR> .
2021-08-23 오후 08:02 <DIR> ..
2021-08-23 오후 08:02 684 manage.py
2021-08-23 오후 08:02 <DIR> mysite
                  1개 파일          684 바이트
                  3개 디렉터리 20,985,503,744 바이트 남음

(base) C:\MyTest\projectsite>python manage.py startapp polls
(base) C:\MyTest\projectsite>
```

프로젝트 뼈대 만들기

애플리케이션 생성

- polls라는 애플리케이션 폴더 확인
- (base) C:\MyTest\projectsite\polls>dir

```
(base) C:\MyTest\projectsite>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest\projectsite 디렉터리

2021-08-23 오후 10:41 <DIR> .
2021-08-23 오후 10:41 <DIR> ..
2021-08-23 오후 08:02      684 manage.py
2021-08-23 오후 10:41 <DIR> mysite
2021-08-23 오후 10:41 <DIR> polls
                1개 파일             684 바이트
                4개 디렉터리  20,989,534,208 바이트 남음
```

```
(base) C:\MyTest\projectsite>cd polls

(base) C:\MyTest\projectsite\polls>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA

C:\MyTest\projectsite\polls 디렉터리

2021-08-23 오후 10:41 <DIR> .
2021-08-23 오후 10:41 <DIR> ..
2021-08-23 오후 10:41      66 admin.py
2021-08-23 오후 10:41     148 apps.py
2021-08-23 오후 10:41 <DIR> migrations
2021-08-23 오후 10:41     60 models.py
2021-08-23 오후 10:41     63 tests.py
2021-08-23 오후 10:41     66 views.py
2021-08-23 오후 10:41      0 __init__.py
                6개 파일             403 바이트
                3개 디렉터리  20,989,468,672 바이트 남음
```

프로젝트 뼈대 만들기

프로젝트 설정 파일 변경

- 프로젝트에 필요한 설정 값들은 **settings.py** 파일에 지정
- settings.py 파일은 프로젝트의 전반적인 사항들을 설정해주는 곳으로, 루트 디렉토리를 포함한 각종 디렉토리의 위치, 로그의 형식, 프로젝트에 포함된 애플리케이션의 이름 등이 지정되어 있음

```
(base) C:\MyTest\projectsite>cd mysite
```

```
(base) C:\MyTest\projectsite\mysite>notepad settings.py
```

프로젝트 뼈대 만들기

프로젝트 설정 파일 변경

- ALLOWED_HOSTS 항목을 적절하게 지정
 - `ALLOWED_HOSTS = ['192.168.35.61', 'localhost', '127.0.0.1']`
- 애플리케이션들은 모두 설정 파일에 등록(polls 애플리케이션도 등록)
 - `INSTALLED_APPS = [~, 'polls.apps.PollsConfig',]`
- 프로젝트에 사용할 데이터베이스 엔진
 - 장고는 디폴트로 SQLite3 데이터베이스 엔진을 사용하도록 설정
- 타임존 지정(기본은 세계표준시(UTC)로 되어 있음. 한국시간을 변경)
 - `#TIME_ZONE = 'UTC'`
 - `TIME_ZONE = 'Asia/Seoul'`

프로젝트 뼈대 만들기

기본 테이블 생성

- migrate 명령은 데이터베이스에 변경사항이 있을 때 반영해주는 명령
- (base) C:\MyTest\projectsite>python manage.py migrate

```
(base) C:\MyTest\projectsite>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

```
(base) C:\MyTest\projectsite>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 421A-9EBA
```

C:\MyTest\projectsite 디렉터리

2021-08-23	오후 11:30	<DIR>	.
2021-08-23	오후 11:30	<DIR>	..
2021-08-23	오후 11:30		131,072 db.sqlite3
2021-08-23	오후 08:02		684 manage.py
2021-08-23	오후 10:41	<DIR>	mysite
2021-08-23	오후 11:30	<DIR>	polls
	2개 파일		131,756 바이트
	4개 디렉터리		20,995,629,056 바이트 남음

프로젝트 뼈대 만들기

작업 확인하기

- 프로젝트의 뼈대에 해당하는 프로젝트 디렉토리, 애플리케이션 디렉토리를 비롯해 관련 파일들 그리고 사용자 및 그룹 테이블을 만들었음
- 확인을 위해서 웹 서버를 실행하고, 그 웹 서버에 접속
- 장고에서는 개발 과정 도중에 현재 상태를 확인해볼 수 있도록 **runserver**라고 하는 간단한 테스트용 웹 서버를 제공

프로젝트 뼈대 만들기

작업 확인하기

- 웹 서버를 실행하기 위해서 다음 명령을 입력합니다. 하나의 창에서 작업 해도 되지만 runserver 용으로 별도의 cmd창을 열어 사용하는 것이 편리
- (base) C:\MyTest\projectsite>python manage.py runserver 127.0.0.1:8000
- (base) C:\MyTest\projectsite>python manage.py runserver
- (base) C:\MyTest\projectsite>python manage.py runserver 0:8000

```
(base) C:\MyTest>cd projectsite

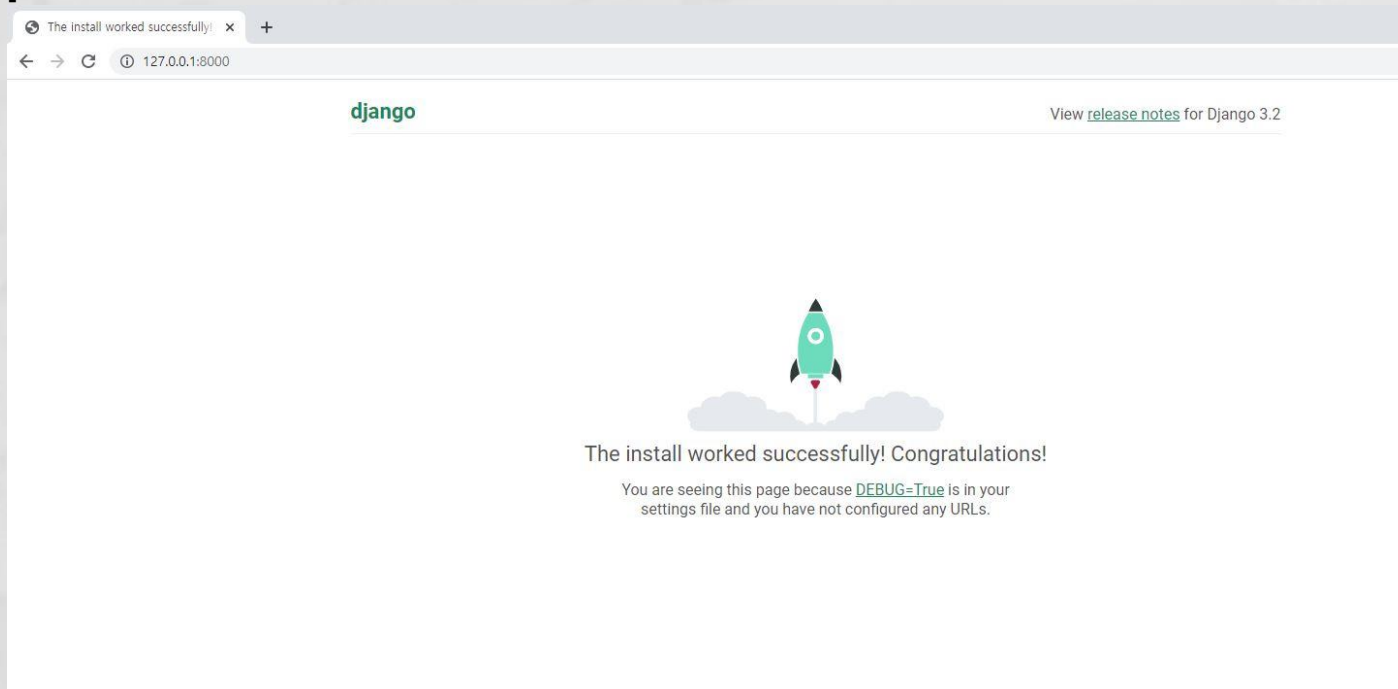
(base) C:\MyTest\projectsite>python manage.py runserver 127.0.0.1:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 23, 2021 - 23:40:17
Django version 3.2.6, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```


프로젝트 뼈대 만들기

작업 확인하기

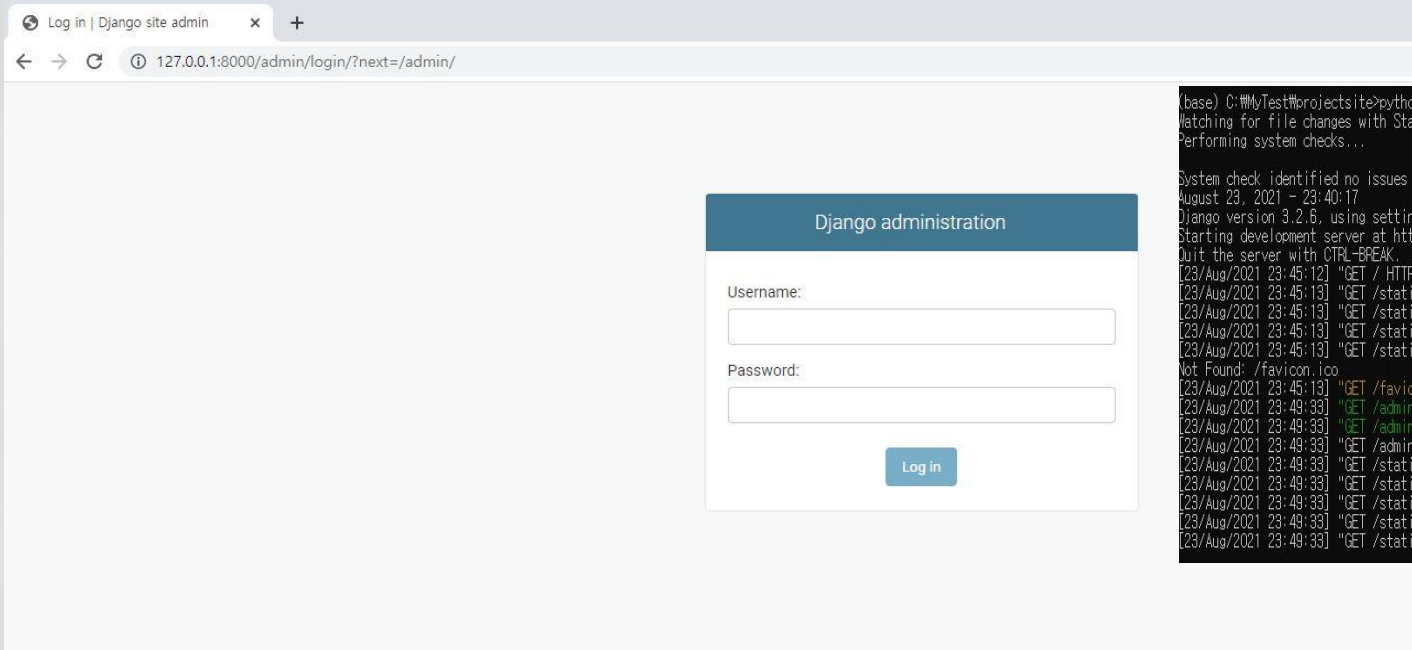
- runserver가 정상적으로 실행되었다면, 웹 브라우저를 열고 IP 주소는 runserver가 동작하는 서버 IP주소를 입력
- `http://127.0.0.1:8000/`



프로젝트 뼈대 만들기

Admin 사이트 접속

- 기본적으로 제공하는 Admin 사이트에 접속해서 테이블이 생성 확인
- 브라우저의 주소창에 IP 주소와 포트번호 동일, URL경로만 /admin 추가
- `http://127.0.0.1:8000/admin`



```
(base) C:\Test\project\site>python manage.py runserver 127.0.0.1:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 23, 2021 - 23:40:17
Django version 3.2.6, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[23/Aug/2021 23:45:12] "GET / HTTP/1.1" 200 10697
[23/Aug/2021 23:45:13] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[23/Aug/2021 23:45:13] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184
[23/Aug/2021 23:45:13] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85878
[23/Aug/2021 23:45:13] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
Not Found: /favicon.ico
[23/Aug/2021 23:45:13] "GET /favicon.ico HTTP/1.1" 404 2110
[23/Aug/2021 23:49:33] "GET /admin HTTP/1.1" 301 0
[23/Aug/2021 23:49:33] "GET /admin/ HTTP/1.1" 302 0
[23/Aug/2021 23:49:33] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 2218
[23/Aug/2021 23:49:33] "GET /static/admin/css/base.css HTTP/1.1" 200 19513
[23/Aug/2021 23:49:33] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2271
[23/Aug/2021 23:49:33] "GET /static/admin/css/login.css HTTP/1.1" 200 939
[23/Aug/2021 23:49:33] "GET /static/admin/css/responsive.css HTTP/1.1" 200 18545
[23/Aug/2021 23:49:33] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 1360
```

프로젝트 뼈대 만들기

Admin 사이트 접속

- 로그인하려면 username, password를 넣어야 되는데 아직 생성하지 않았음
- Admin 사이트에 로그인하기 위한 관리자(슈퍼유저)를 생성
- (base) C:\MyTest\projectsite>python manage.py createsuperuser

```
(base) C:\MyTest\projectsite>python manage.py createsuperuser
Username (leave blank to use 'bbobbee'): smcpacs
Email address: smcpacs@naver.com
Password:
Password (again):
```

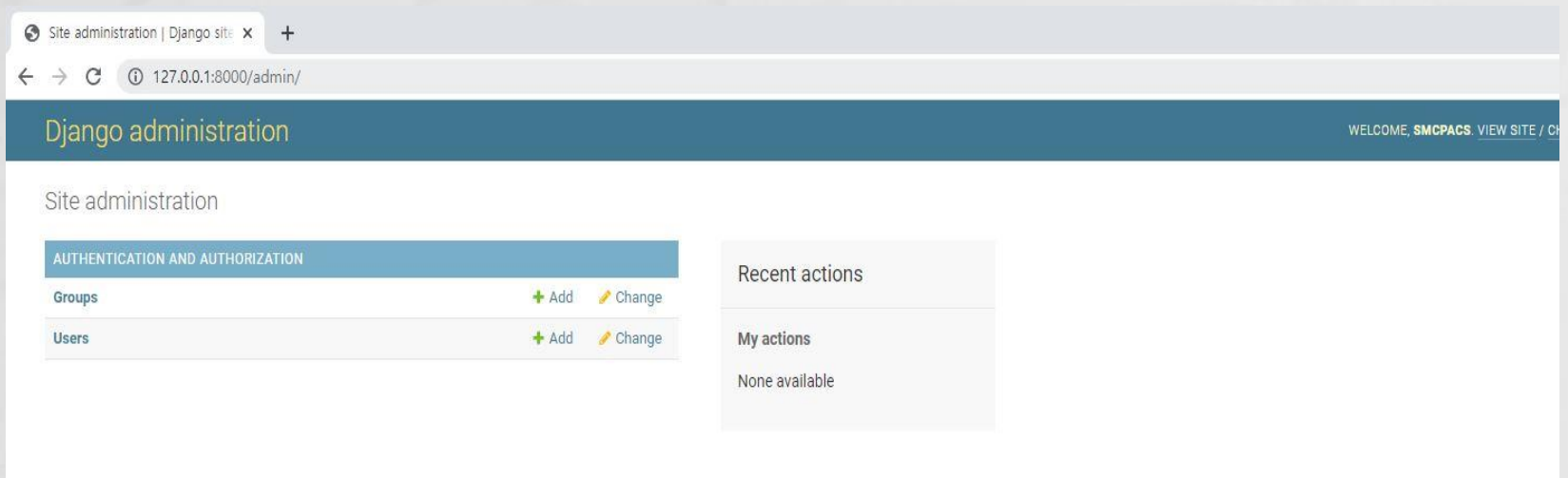
```
Superuser created successfully.
```

```
(base) C:\MyTest\projectsite>
```

프로젝트 뼈대 만들기

Admin 사이트 접속

- `http://127.0.0.1:8000/admin`
- 로그인 창에서 `username`, `password` 입력



프로젝트 뼈대 만들기

Admin 사이트 접속

○ 사용자 확인

Select user to change | Django

127.0.0.1:8000/admin/auth/user/

Django administration

WELCOME, **SMCPACS**. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Authentication and Authorization > Users

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

Select user to change

Search

Action: [-----] Go 0 of 1 selected

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

1 user

FILTER

- By staff status
 - All
 - Yes
 - No
- By superuser status
 - All
 - Yes
 - No
- By active
 - All
 - Yes
 - No

프로젝트 뼈대 만들기

Admin 사이트 접속

- Admin 사이트에서 Users와 Groups 테이블을 포함하여 새롭게 만들 테이블에 대한 데이터의 입력, 변경, 삭제 등의 작업을 할 수 있음
- Admin 화면에서 기본적으로 Users와 Groups 테이블이 보이는 것은 이미 settings.py 파일에 django.contrib.auth 애플리케이션이 등록되어 있기 때문
- 즉 장고에서 기본으로 제공하는 auth 앱에 Users와 Groups 테이블이 미리 정의

프로젝트 뼈대 만들기

골격 생성

- 애플리케이션을 MVT 패턴 방식으로 개발할 수 있도록 골격 생성
- (base) C:\MyTest>tree /F projectsite

```
(base) C:\MyTest>tree /F projectsite
폴더 PATH의 목록입니다.
블록 일련 번호가 00000034 421A:9EBA입니다.
C:\MYTEST\PROJECTSITE
db.sqlite3
manage.py

mysite
  asgi.py
  settings.py
  urls.py
  wsgi.py
  __init__.py
  __pycache__
    settings.cpython-38.pyc
    urls.cpython-38.pyc
    wsgi.cpython-38.pyc
    __init__.cpython-38.pyc

polls
  admin.py
  apps.py
  models.py
  tests.py
  views.py
  __init__.py

migrations
  __init__.py
  __pycache__
    __init__.cpython-38.pyc

__pycache__
  admin.cpython-38.pyc
  apps.cpython-38.pyc
  models.cpython-38.pyc
  __init__.cpython-38.pyc
```

정리

정리

- 일반적인 특징
- 장고에서의 애플리케이션