

---

# Django 웹 프레임워크

# 장고 프로그램 골격 만들기

---

## ❖ 가상 환경 만들기

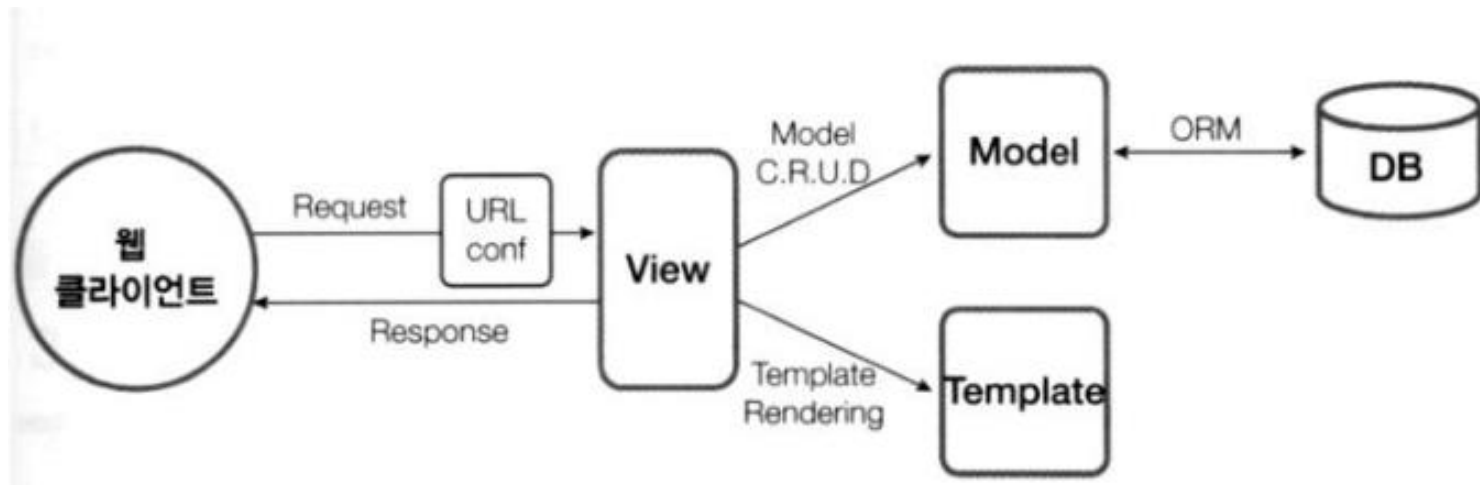
- `conda create -n iot python=3.7`
- `conda activate iot`

## ❖ 장고 설치

- `(iot) pip install Django`

# 장고 프로그램 골격 만들기

## ❖ 장고 패턴



# 장고 프로그램 골격 만들기

---

## ❖ 프로젝트 뼈대 만들기

- 프로젝트 만들기
- (iot) `django-admin startproject mysite`
- (iot) `ren mysite web_app1`
- (iot) `cd web_app1`

```
web_app1
├── manage.py
└── mysite
    ├── asgi.py
    ├── settings.py
    ├── urls.py
    ├── wsgi.py
    └── __init__.py
```

- web\_app1 프로젝트 디렉토리를 IDE로 오픈

# 장고 프로그램 골격 만들기

---

## ❖ polls 앱 만들기

- (iot) python manage.py startapp polls

```
web_app1
├── manage.py
├── mysite
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   └── __init__.py
└── polls
    ├── admin.py
    ├── apps.py
    ├── models.py
    ├── tests.py
    ├── views.py
    ├── __init__.py
    └── migrations
        └── __init__.py
```

# 장고 프로그램 골격 만들기

---

## ❖ polls 앱 만들기

- mysite/settings.py

```
# Application definition
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'polls',
]

:

LANGUAGE_CODE = 'ko-kr'
TIME_ZONE = 'Asia/Seoul'

:
```

# 장고 프로그램 골격 만들기

---

## ❖ 프로젝트 초기화

- 기반 데이터베이스 테이블(계정 관리) 만들기
- (iot) `python manage.py migrate`
- 프로젝트 디렉토리에 `db.sqlite3` 파일 생성됨

## ❖ 웹 서버 기동

- (iot) `python manage.py runserver`
- 브라우저로 접속
  - `http://127.0.0.1:8000/`

# 장고 프로그램 골격 만들기

---

## ❖ 수퍼 유저 만들기

- (iot) python manage.py createsuperuser

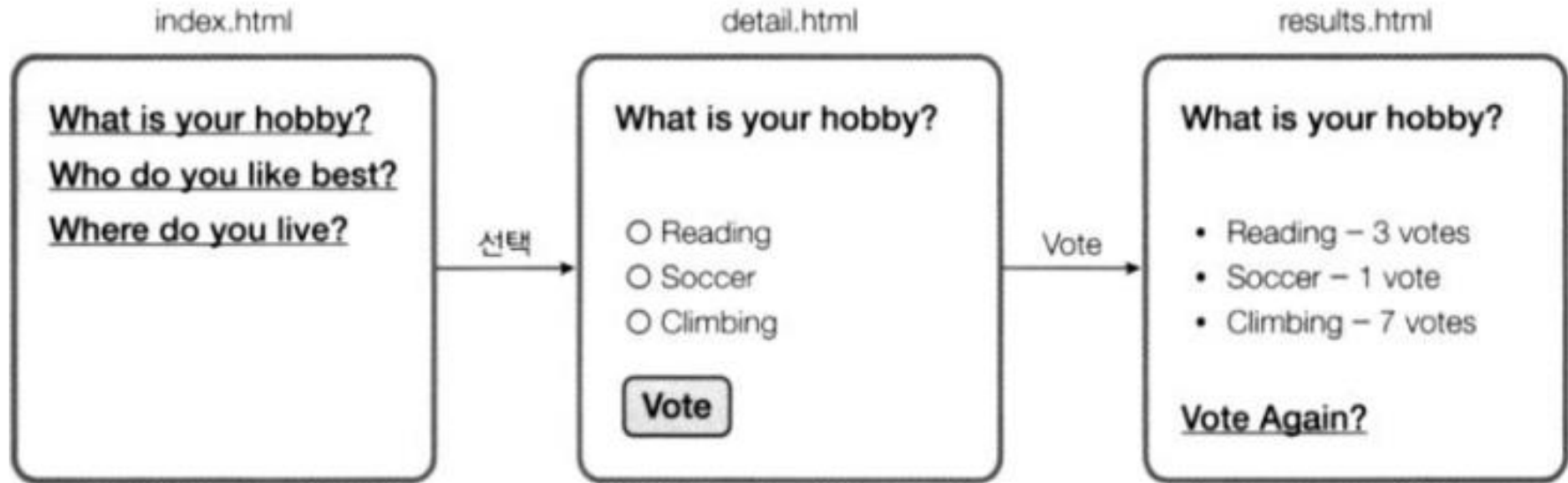
```
사용자 이름 (leave blank to use 'i'): admin
이메일 주소: admin@test.com
Password:
Password (again):
비밀번호가 너무 짧습니다. 최소 8 문자를 포함해야 합니다.
비밀번호가 너무 일상적인 단어입니다.
비밀번호가 전부 숫자로 되어 있습니다.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

- 수퍼유저로 관리자 페이지 로그인
  - <http://127.0.0.1/admin>



# Polls 앱

## ❖ UI 설계



템플릿	설명
index.html	최근에 실시한 질문의 리스트를 보여 줌
detail.html	하나의 질문에 대해 투표할 수 있도록 답변 항목을 폼으로 보여 줌
results.html	질문에 따른 투표 결과를 보여 줌

# Polls 앱

---

## ❖ 모델 설계

- 모델 클래스 정의
  - 테이블 정의에 해당
    - Question : 질문을 저장하는 테이블
    - Choice : 질문 별로 선택용 답변 항목을 저장하는 테이블

# Polls 앱

## ❖ 모델 설계

### ○ Question

컬럼명	타입	제약조건	설명
id	integer	NotNull, PK, AutoIncrement	Primary Key
question	varchar(200)	NotNull	질문 문장
pub_date	datetime	NotNull	질문 생성 시간

### ○ Choice 테이블

컬럼명	타입	제약조건	설명
id	integer	NotNull, PK, AutoIncrement	Primary Key
choice_text	varchar(200)	NotNull	답변 항목 문구
votes	integer	NotNull	투표 카운트
question_id	integer	NotNull, FK(Question.id), Index	Foreign Key

# Polls 앱

---

## ❖ 데이터베이스 설정

- mysite/settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

# Polls 앱

---

## ❖ 모델 정의

- polls/modles.py

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date      = models.DateTimeField('date published')

    def __str__(self):
        return self.question_text

class Choice(models.Model):
    question      = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text   = models.CharField(max_length=200)
    votes         = models.IntegerField(default=0)

    def __str__(self):
        return self.choice_text
```

# Polls 앱

---

## ❖ 모델 정의

- id 컬럼을 생략한 경우 자동 생성
- `models.CharField`
  - `varchar` 타입
  - `max_length` 속성 필수
- `models.IntegerField`
  - `integer` 타입
- `models.DateTimeField`
  - `Datetime` 타입
- `models.ForeignKey`
  - 참조 테이블에 해당하는 모델 클래스 지정
  - 지정한 참조 모델의 PK와 동일타입
  - `on_delete=models.CASCADE`으로 삭제 시 행동 지정
- 공통
  - 첫 번째 인자로 문자열을 제시한 경우 → 해당 컬럼의 설명 문구

# Polls 앱

---

## ❖ Admin 사이트에 모델 등록

- polls/admin.py

```
from django.contrib import admin  
from polls.models import Question, Choice
```

```
admin.site.register(Question)  
admin.site.register(Choice)
```

# Polls 앱

---

## ❖ 데이터베이스 반영하기

- `(iot) python manage.py makemigrations`
- `(iot) python manage.py migrate`
  
- 데이터베이스에 실제 테이블이 생성됨



# Polls 앱

---

## ❖ 확인 및 데이터 등록

- <http://127.0.0.1:8000/admin>
- Question과 Choice 테이블에 샘플 데이터 추가
  - What is your hobby?
    - Reading
    - Soccer
    - Climbing
  - Where do you live?
    - Seoul
    - Suwon
    - Busan
    - Jeju

## ❖ 확인 및 데이터 등록

### Django 관리

#### 사이트 관리

##### POLLS

###### Choices

+ 추가    ✎ 변경

###### Questions

+ 추가    ✎ 변경

##### 인증 및 권한

###### 그룹

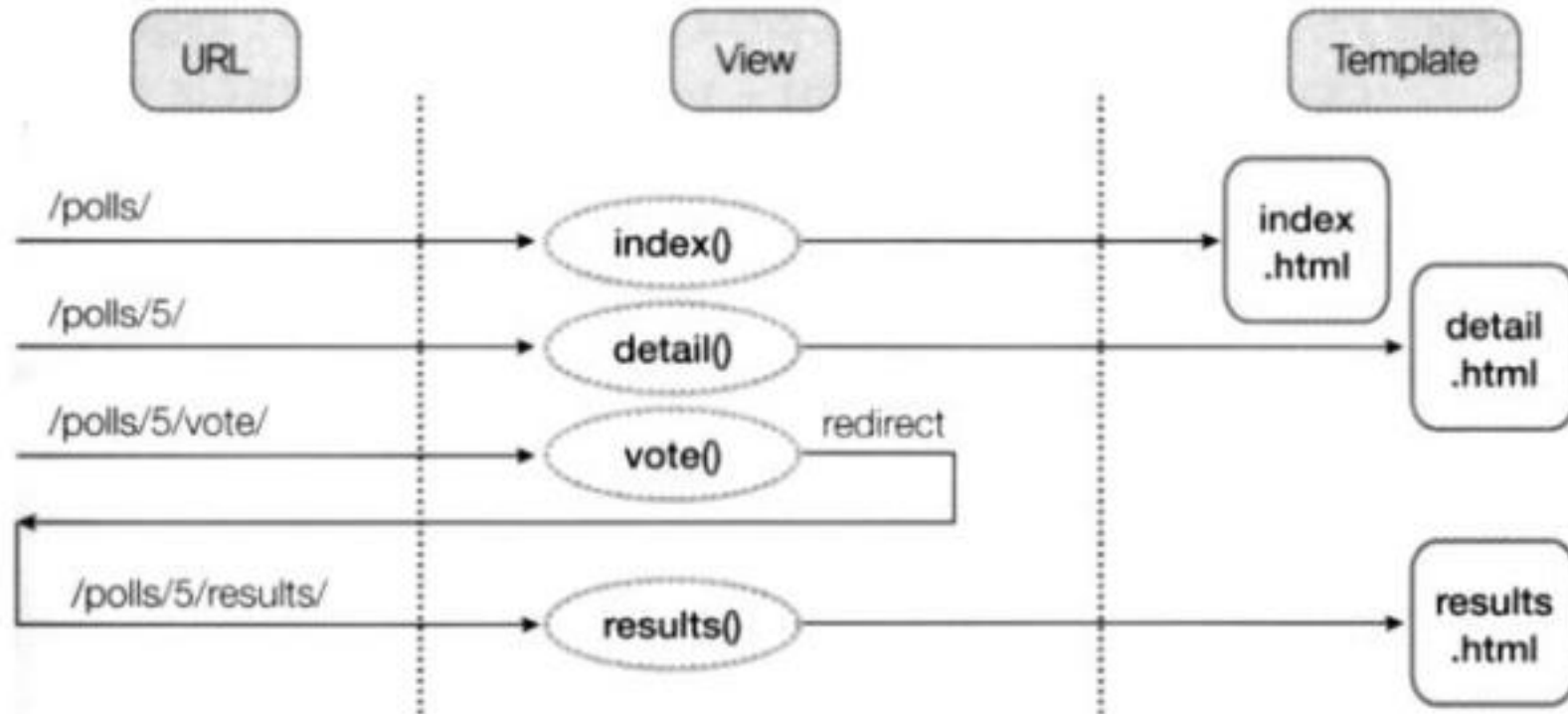
+ 추가    ✎ 변경

###### 사용자(들)

+ 추가    ✎ 변경

# Polls 앱

## ❖ View/Template



# Polls 앱

## ❖ URL 설계

URL 패턴	뷰 이름	뷰가 처리하는 내용
/polls/	index()	index.html 템플릿을 보여 줌
/polls/5/	detail()	detail.html 템플릿을 보여 줌
/polls/5/vote	vote()	detail.html에 있는 폼을 POST 방식으로 처리 처리 후 리다이렉트
/polls/5/results	result()	results.html 템플릿을 보여 줌
/admin/	(장고 기본 기능)	Admin 사이트를 보여줌

# Polls 앱

## ❖ URLconf 구성

- polls/urls.py

```
from django.urls import path
from polls import views
```

```
app_name = 'polls'      # polls 앱의 namespace명
```

```
urlpatterns = [
    # /polls/
    path('', views.index, name='index'),
    # /polls/5/
    path('<int:question_id>/', views.detail, name='detail'),
    # /polls/5/results/
    path('<int:question_id>/results/', views.results, name='results'),
    # /polls/5/vote/
    path('<int:question_id>/vote/', views.vote, name='vote'),
]
```

# Polls 앱

---

## ❖ URLconf 구성

- mysite/urls.py

```
from django.contrib import admin
from django.urls import path, include

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

# Polls 앱

---

## ❖ View 정의

- polls/views.py

```
from django.shortcuts import render
from polls.models import Choice, Question

def index(request):
    pass

def detail(request, question_id):
    pass

def vote(request, question_id):
    pass

def results(request, question_id):
    pass
```

# Polls 앱

## ❖ View 정의 - 목록 보기

- o polls/views.py

```
from django.shortcuts import render
from polls.models import Choice, Question

def index(request):
    latest_question_list = Question.objects.all().order_by('-pub_date')[:5]

    context = {'latest_question_list': latest_question_list}

    return render(request, 'polls/index.html', context)
```

Question	모델 클래스
.objects	Manager 클래스
.all()	전체 행 추출
.order_by('-pub_date')	pub_date 컬럼을 기준으로 내림차순 정렬
[:5]	앞에 5개 행만 선택



# Polls 앱

---

## ❖ View 정의 - 목록 보기

- polls/templates/polls/index.html

```
{% if latest_question_list %}
    <ul>
    {% for question in latest_question_list %}
        <li><a href="/polls/{{question.id}}/">
            {{ question.question_text }}</a></li>
    {% endfor %}
    </ul>
{% else %}
    <p>No polls are available.</p>
{% endif %}
```

# Polls 앱

## ❖ View 정의 - 단일 항목 보기

○ polls/views.py

```
from django.shortcuts import get_object_or_404, render

:

def detail(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/detail.html', {'question': question})
```

- `path('<int:question_id>/', views.detail, name='detail')`에서 정의한 경로 변수 `question_id`와 같은 이름으로 매개변수 이름 지정
- `get_object_or_404(Question, pk=question_id)`
  - 지정한 모델에서 primary key에 해당하는 `question_id` 값으로 검색
  - 존재하면 모델 인스턴스 리턴
  - 존재하지 않으면 404 에러 발생

# Polls 앱

## ❖ View 정의 - 단일 항목 보기

- o polls/templates/polls/detail.html

```
<h1>{{ question.question_text }}</h1>

{% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}

<form action="{% url 'polls:vote' question.id %}" method="post">
    {% csrf_token %}
    {% for choice in question.choice_set.all %}
    <input type="radio" name="choice" id="choice{{ forloop.counter }}"
        value="{{ choice.id }}" />
    <label for="choice{{ forloop.counter }}">
        {{ choice.choice_text }}</label><br />
    {% endfor %}
    <input type="submit" value="Vote" />
</form>
```

# Polls 앱

## ❖ View 정의 - 투표처리/리다이렉트

- o polls/views.py

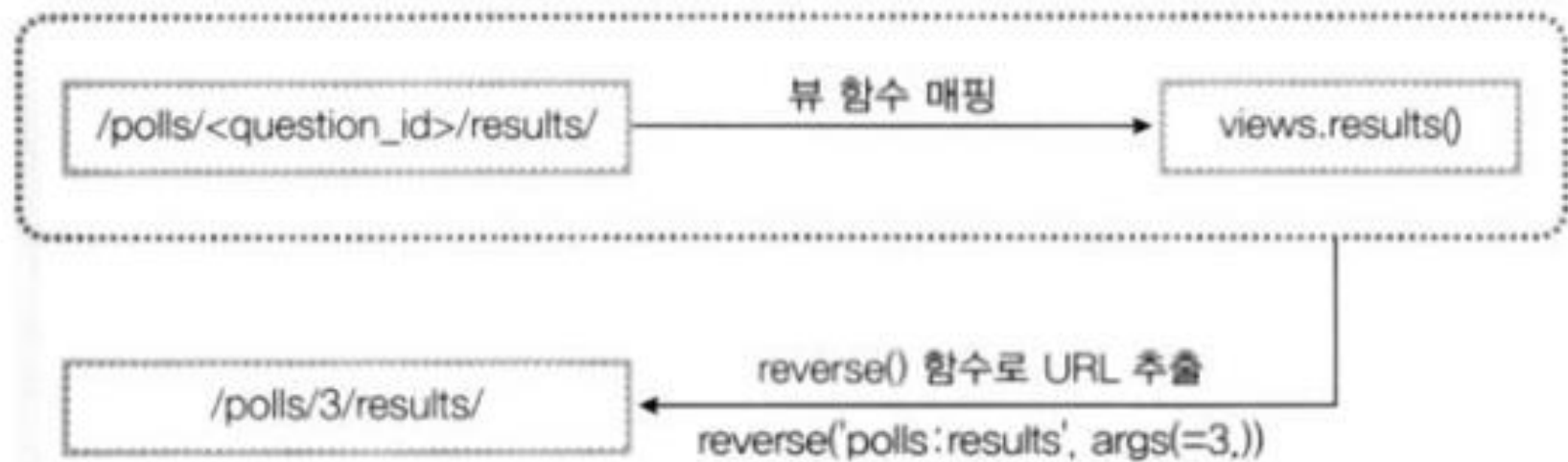
```
from django.shortcuts import get_object_or_404, render
from django.http import HttpResponseRedirect
from django.urls import reverse
:

def vote(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        return render(request, 'polls/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        return HttpResponseRedirect(
            reverse('polls:results', args=(question.id,)))
```

# Polls 앱

## ❖ reverse() 함수의 동작 원리

URLconf에 정의된 URL 패턴명 = polls:results



# Polls 앱

---

## ❖ View 정의 - 결과 보기

- polls/views.py

```
:
```

```
def results(request, question_id):  
    question = get_object_or_404(Question, pk=question_id)  
  
    return render(request, 'polls/results.html', {'question': question})
```

# Polls 앱

---

## ❖ View 정의 - 단일 항목 보기

- polls/templates/polls/results.html

```
<h1>{{ question.question_text }}</h1>
```

```
<ul>
```

```
    {% for choice in question.choice_set.all %}
```

```
    <li>{{ choice.choice_text }} --
```

```
        {{ choice.votes }} vote{{ choice.votes|pluralize }}</li>
```

```
    {% endfor %}
```

```
</ul>
```

```
<a href="{% url 'polls:detail' question.id %}">Vote again?</a>
```