SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Django 02
Template & URLs

SAMSUNG SW ACADEMY FOR YOUTH

• INDEX

- Template System
  - Django Template system
  - Django Template Language
- 템플릿 상속
  - 상속 관련 DTL 태그
- HTML form
  - 요청과 응답
  - form 활용

SAMSUNG SW ACADEMY FOR YOUTH

• INDEX

- Django URLs
  - Variable Routing
  - App과 URL
- URL 이름 지정
  - Naming URL patterns
  - DTL URL tag
- URL 이름 공간
  - app_name 속성
- 참고
  - 추가 템플릿 경로
  - DTL 주의사항
  - Trailing Slashes

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Template System

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Django Template system

Confidential

Django Template system

데이터 표현을 제어하면서, 표현과 관련된 부분을 담당

Confidential

HTML의 콘텐츠를 변수 값에 따라 변경하기 (1/2)

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

<body>
  <h1>Hello, django!</h1>
</body>
```

Hello, django!

Confidential

HTML의 콘텐츠를 변수 값에 따라 변경하기 (2/2)

```python
# articles/views.py

def index(request):
    context = {
        'name': 'Jane',
    }
    return render(request, 'articles/index.html', context)
```

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

<body>
  <h1>Hello, {{ name }}</h1>
</body>
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Django Template Language

Confidential

Django Template Language
(DTL)

Template에서 조건, 반복, 변수 등의
프로그래밍적 기능을 제공하는 시스템

Confidential

DTL Syntax

1. Variable
2. Filters
3. Tags
4. Comments

Confidential

1. Variable

- `render` 함수의 세번째 인자로 딕셔너리 데이터를 사용
- 딕셔너리 key에 해당하는 문자열이 template에서 사용 가능한 변수명이 됨
- dot('.')를 사용하여 변수 속성에 접근할 수 있음

```
{{ variable }}
```

```
{{ variable.attribute }}
```

Confidential

2. Filters

- 표시할 변수를 수정할 때 사용 (변수 + '¦' + 필터)
- chained(연결)이 가능하며 일부 필터는 인자를 받기도 함
- 약 60개의 built-in template filters를 제공

```
{{ variable¦filter }}
```

```
{{ name¦truncatewords:30 }}
```

Confidential

3. Tags

- 반복 또는 논리를 수행하여 제어 흐름을 만듦
- 일부 태그는 시작과 종료 태그가 필요
- 약 24개의 built-in template tags를 제공

```
{% tag %}
```

```
{% if %} {% endif %}
```

Confidential

4. Comments

- DTL에서의 주석

```
<h1>Hello, {# name #}</h1>
```

```
{% comment %}
  …
{% endcomment %}
```

Confidential

DTL 예시 (1/2)

국밥 메뉴는 2글자 입니다.

메뉴판

- 국밥
- 국수
- 카레
- 탕수육

아직 메뉴가 남았습니다.

Confidential

DTL 예시 (2/2)

```python
# urls.py

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index/', views.index),
    path('dinner/', views.dinner),
]
```

```python
# views.py

import random

def dinner(request):
    foods = ['국밥', '국수', '카레', '탕수육',]
    picked = random.choice(foods)
    context = {
        'foods': foods,
        'picked': picked,
    }
    return render(request, 'articles/dinner.html', context)
```

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

<p>{{ picked }} 메뉴는 {{ picked|length }}글자 입니다.</p>
<h2>메뉴판</h2>
<ul>
  {% for food in foods %}
    <li>{{ food }}</li>
  {% endfor %}
</ul>

{% if foods|length == 0 %}
  <p>메뉴가 소진 되었습니다.</p>
{% else %}
  <p>아직 메뉴가 남았습니다.</p>
{% endif %}
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG
SW
ACADEMY
FOR
YOUTH

함께가요 미래로!
Enabling People

템플릿 상속

Confidential

기본 템플릿 구조의 한계

- 만약 모든 템플릿에 bootstrap을 적용하려면?

➢ 모든 템플릿에 bootstrap CDN을 작성해야 할까?

Confidential

템플릿 상속

Template inheritance

페이지의 공통요소를 포함하고,
하위 템플릿이 재정의 할 수 있는 공간을 정의하는
기본 'skeleton' 템플릿을 작성하여 상속 구조를 구축

Confidential

템플릿 상속

Template inheritance

1 페이지의 공통요소를 포함하고,
2 하위 템플릿이 재정의 할 수 있는 공간을 정의하는
기본 'skeleton' 템플릿을 작성하여 상속 구조를 구축

Confidential

상속 구조 만들기 (1/2)

skeleton 역할을 하게 되는 상위 템플릿(base.html) 작성

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

<!DOCTYPE html>
<html lang="en">
<head>
  …
  {% comment %} bootstrap CDN 생략 {% endcomment %}
</head>
<body>
  {% block content %}
  {% endblock content %}
  {% comment %} bootstrap CDN 생략 {% endcomment %}
</body>
</html>
```

Confidential

상속 구조 만들기 (2/2)

기존 하위 템플릿의 변화

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

{% extends 'articles/base.html' %}

{% block content %}
  <h1>Hello, {{ name }}</h1>
{% endblock content %}
```

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

{% extends 'articles/base.html' %}

{% block content %}
  <p>{{ picked }} 메뉴는 {{ picked|length }}글자 입니다.</p>

  <h2>메뉴판</h2>
  <ul>
    {% for food in foods %}
      <li>{{ food }}</li>
    {% endfor %}
  </ul>

  {% if foods|length == 0 %}
    <p>메뉴가 소진되었습니다.</p>
  {% else %}
    <p>아직 메뉴가 남았습니다.</p>
  {% endif %}
{% endblock content %}
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

상속 관련 DTL 태그

Confidential

‘extends’ tag

```
{% extends ‘path’ %}
```

자식(하위)템플릿이 부모 템플릿을 확장한다는 것을 알림

❖ 반드시 자식 템플릿 최상단에 작성되어야 함 (2개 이상 사용 불가)

Confidential

‘block’ tag

```
{% block name %}{% endblock name %}
```

하위 템플릿에서 재정의 할 수 있는 블록을 정의
(상위 템플릿에 작성하며
하위 템플릿이 작성할 수 있는 공간을 지정하는 것)

Confidential

하위 템플릿이 재정의 할 수 있는 block 영역

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

<!DOCTYPE html>
<html lang="en">
<head>
  …
  {% comment %} bootstrap CDN 생략 {% endcomment %}
</head>
<body>
  {% block content %}
  {% endblock content %}
  {% comment %} bootstrap CDN 생략 {% endcomment %}
</body>
</html>
```

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

{% extends 'articles/base.html' %}

{% block content %}
  <h1>Hello, {{ name }}</h1>
{% endblock content %}
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

HTML form

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

요청과 응답

Confidential

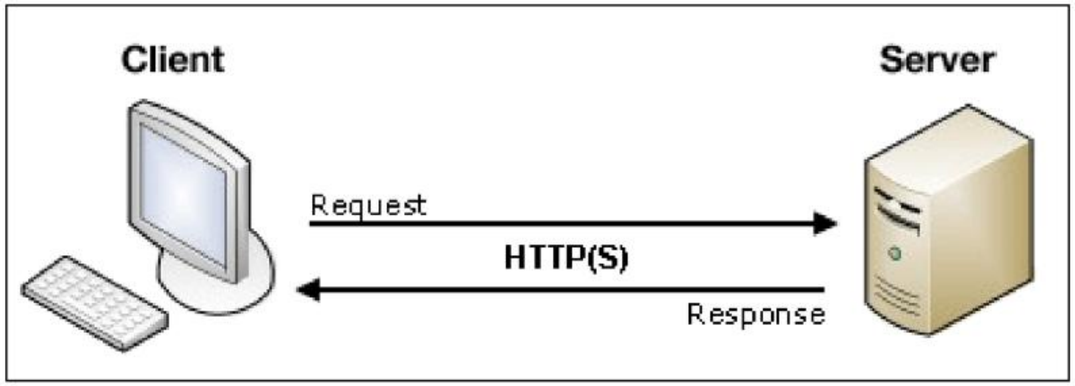
데이터를 보내고 가져오기

Sending and Retrieving form data

HTML ‘form’ element를 통해
사용자와 애플리케이션 간의 상호작용 이해하기

Confidential

HTML ‘form’은
HTTP 요청을 서버에 보내는 가장 편리한 방법



Confidential

HTML ‘form’은
HTTP 요청을 서버에 보내는 가장 편리한 방법

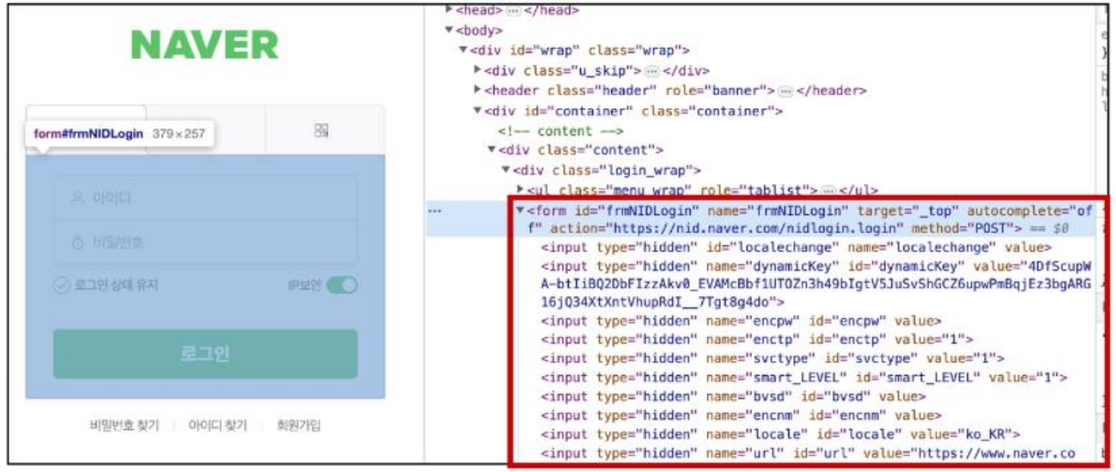
아이디 :
패스워드 :
로그인

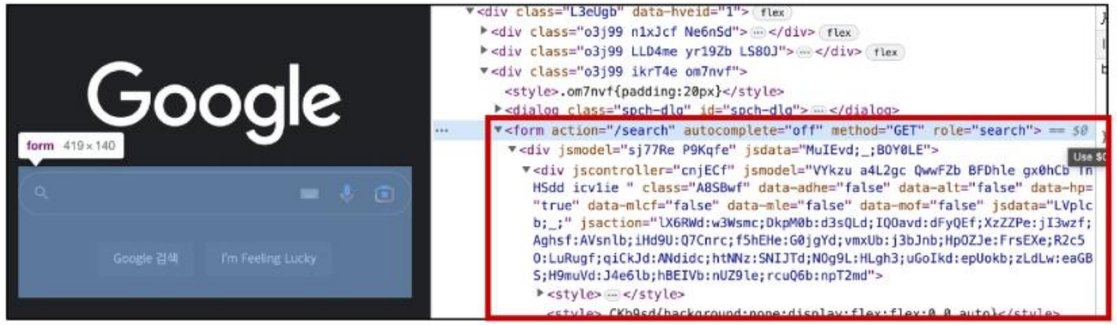
```
<form action="#“ method="GET">
  <div>
    <label for="name">아이디 : </label>
    <input type="text" id="name">
  </div>
  <div>
    <label for="password">패스워드 : </label>
    <input type="password" name="password" id="password">
  </div>
  <input type="submit" value="로그인">
</form>
```

Confidential

실제 웹 서비스에서 form이 사용되는 예시

네이버 & 구글의 로그인 화면에서 사용하는 HTML form 요소





Confidential

‘form’ element

사용자로부터 할당된 데이터를 서버로 전송

- 웹에서 사용자 정보를 입력하는 여러 방식 (text, password, checkbox 등)을 제공

Confidential

fake Naver 실습 (1/4)

```python
# urls.py

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index/', views.index),
    path('dinner/', views.dinner),
    path('search/', views.search),
]
```

```python
# views.py

def search(request):
    return render(request, 'articles/search.html')
```

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

{% extends 'articles/base.html' %}

{% block content %}
  <form action="#" method="GET">
    <label for="message">검색어</label>
    <input type="text" name=" message" id="message">
    <input type="submit" value="submit">
  </form>
{% endblock content %}
```

Confidential

fake Naver 실습 (2/4)

input에 hello를 입력하고 제출 버튼을 누른 후 브라우저의 URL 변화 확인

http://127.0.0.1:8000/search/?message=hello

```
{% extends 'articles/base.html' %}

{% block content %}
  <form action="#" method="GET">
    <label for="message">검색어</label>
    <input type="text" name="message" id="message">
    <input type="submit" value="submit">
  </form>
{% endblock content %}
```

input에 입력한 데이터

Confidential

fake Naver 실습 (3/4)

실제 Naver에서 검색 후 URL 확인

input의 name 속성

https://search.naver.com/search.naver?query=hello

목적지 URL

input에 입력한 데이터

Confidential

fake Naver 실습 (4/4)

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

{% extends 'articles/base.html' %}

{% block content %}
  <form action="https://search.naver.com/search.naver/" method="GET">
    <label for="message">검색어</label>
    <input type="text" name="query" id="message">
    <input type="submit" value="submit">
  </form>
{% endblock content %}
```

Confidential

‘action’ & ‘method’

form의 핵심 속성 2가지

“데이터를 어디(action)로 어떤 방식(method)으로 요청할지”

Confidential

action과 method

- action
  - 입력 데이터가 전송될 URL을 지정 (목적지)
  - 만약 이 속성을 지정하지 않으면 데이터는 현재 form이 있는 페이지의 URL로 보내짐
- method
  - 데이터를 어떤 방식으로 보낼 것인지 정의
  - 데이터의 HTTP request methods (GET, POST)를 지정

Confidential

‘input’ element

사용자의 데이터를 입력 받을 수 있는 요소
(type 속성 값에 따라 다양한 유형의 입력 데이터를 받음)

- 핵심 속성 - ‘name’

Confidential

‘name’ attribute

- input의 핵심 속성
- 사용자가 입력한 데이터에 붙이는 이름(key)
- 데이터를 제출했을 때 서버는 name 속성에 설정된 값을 통해서만 사용자가 입력한 데이터에 접근할 수 있음

Confidential

Query String Parameters

- 사용자의 입력 데이터를 URL 주소에 파라미터를 통해 서버로 보내는 방법
- 문자열은 앰퍼샌드('&')로 연결된 key=value 쌍으로 구성되며, 기본 URL과는 물음표('?')로 구분됨
- 예시
  - http://host:port/path?key=value&key=value

SAMSUNG SW ACADEMY FOR YOUTH

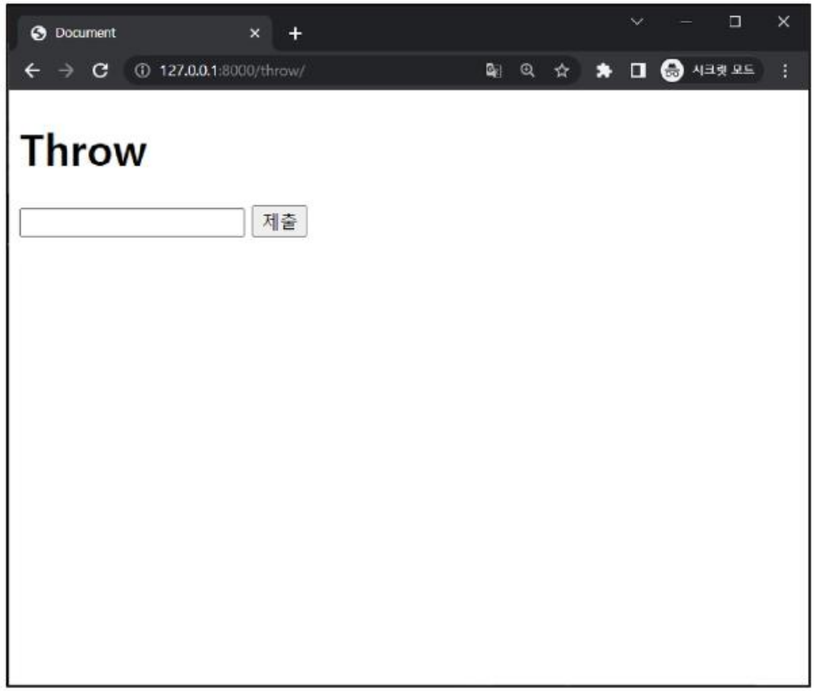
함께가요 미래로!
Enabling People

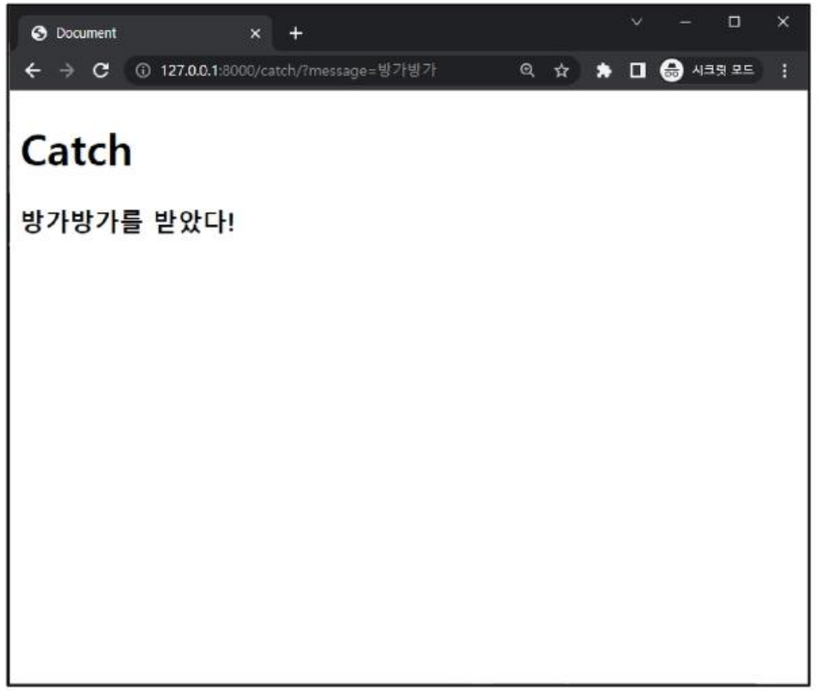
form 활용

Confidential

사용자 입력 데이터를 받아 그대로 출력하는 서버 만들기

view 함수는 몇 개가 필요할까?





Confidential

1. throw 로직 작성

```python
# urls.py

urlpatterns = [
    path('throw/', views.throw),
]
```

```python
# views.py

def throw(request):
    return render(request, 'articles/throw.html')
```

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

{% extends 'articles/base.html' %}

{% block content %}
  <h1>Throw</h1>
  <form action="/catch/" method="GET">
    <input type="text" id="message" name="message">
    <input type="submit">
  </form>
{% endblock content %}
```

Confidential

2. catch 로직 작성

throw 페이지에서 요청한 사용자 입력 데이터는 어떻게 가져와야 할까?

```
# urls.py

urlpatterns = [
    path('catch/', views.catch),
]
```

```
# views.py

def catch(request):
    context = ???
    return render(request, 'articles/catch.html')
```

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

{% extends 'articles/base.html' %}

{% block content %}
  <h1>Catch</h1>
  <h3>{{ ??? }}를 받았습니다!</h3>
{% endblock content %}
```

Confidential

HTTP request 객체

---

form으로 전송한 데이터 뿐만 아니라
Django로 들어오는 모든 요청 관련 데이터가 담겨 있음
(view 함수의 첫번째 인자로 전달됨)

Confidential

request 객체 살펴보기

```python
# views.py

def catch(request):
    print(request)
    print(type(request))
    print(dir(request))
    print(request.GET)
    print(request.GET.get('message'))
    return render(request, 'articles/catch.html')
```

```
<WSGIRequest: GET '/catch/?message=%EC%95%88%EB%85%95%21'>

<class 'django.core.handlers.wsgi.WSGIRequest'>

['COOKIES', 'FILES', 'GET', 'META', 'POST', '__class__',
… 생략… 'parse_file_upload', 'path', 'path_info', 'read',
'readline', 'readlines', 'resolver_match', 'scheme',
'session', 'upload_handlers', 'user']

<QueryDict: {'message': ['안녕!']}>

안녕!
```

출력 결과

Confidential

request 객체에서 form 데이터 추출

```
request.GET.get('message')
```

```
<QueryDict: {'message': ['안녕!']}>
```

딕셔너리의 get 메서드를 사용해
키의 값을 조회

Confidential

3. catch 로직 마무리

```python
# views.py

def catch(request):
    message = request.GET.get('message')
    context = {
        'message': message,
    }
    return render(request, 'articles/catch.html', context)
```

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

{% extends 'articles/base.html' %}

{% block content %}
  <h1>Catch</h1>
  <h3>{{ message }}를 받았습니다!</h3>
{% endblock content %}
```

Confidential

throw - catch 간 요청과 응답 정리 (1/2)

1. `throw/` 로 요청 (throw 페이지를 줘!)

5. 응답 객체 전달

6. 응답 객체 해석 후 화면 출력

2. `throw/` 문자열과 일치하는 `urls.py`의 path 함수 호출
3. `throw` view 함수 호출
4. `throw` view 함수가 응답 객체를 반환

Confidential

throw - catch 간 요청과 응답 정리 (2/2)

2. catch/ 로 요청 (+ 사용자 입력 데이터와 함께)

6. 응답 객체 전달

7. 응답 객체 해석 후 화면 출력

1. throw 페이지에서 form 데이터 작성 후 제출
   (form 요소의 action 속성 값으로 요청)

3. catch/ 문자열과 일치하는
   urls.py의 path 함수 호출
4. catch view 함수 호출
5. catch view 함수에서 사용자가 보낸
   form 데이터 추출 후 응답 객체를 반환

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

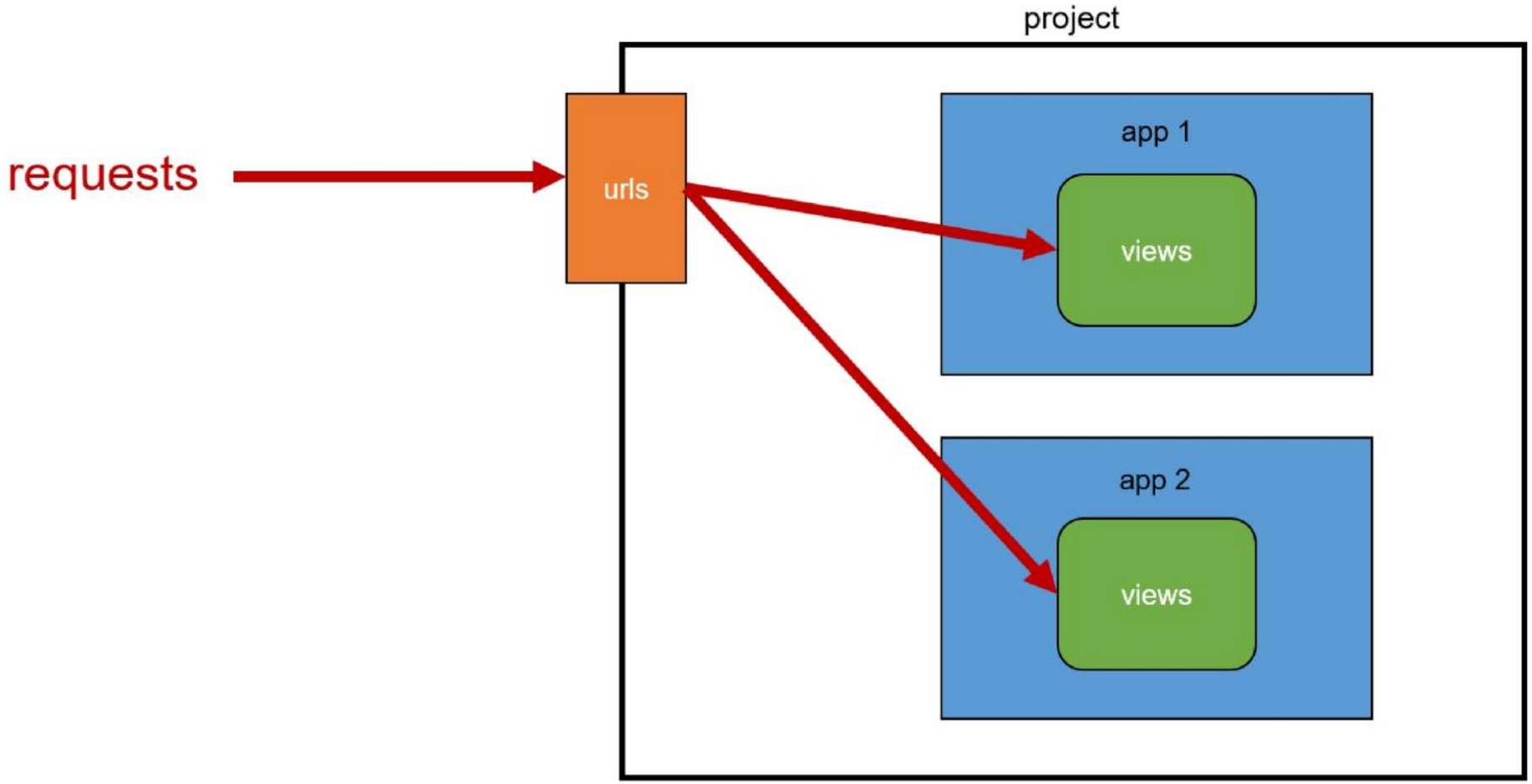
SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Django URLs

Confidential

요청과 응답에서 Django URLs의 역할



Confidential

URL dispatcher

(운항 관리자, 분배기)

URL 패턴을 정의하고

해당 패턴이 일치하는 요청을 처리할 view 함수를 연결(매핑)

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Variable Routing

Confidential

현재 URL 관리의 문제점

템플릿의 많은 부분이 중복되고, URL의 일부만 변경되는 상황이라면
계속해서 비슷한 URL과 템플릿을 작성해 나가야 할까?

```
urlpatterns = [
    path('articles/1/', ...),
    path('articles/2/', ...),
    path('articles/3/', ...),
    path('articles/4/', ...),
    path('articles/5/', ...),
    ...,
]
```

Confidential

Variable Routing

URL 일부에 변수를 포함시키는 것
(변수는 view 함수의 인자로 전달 할 수 있음)

Confidential

Variable routing 작성법

<path_converter:variable_name>

```
path('articles/<int:num>/', views.detail)
path('hello/<str:name>/', views.greeting)
```

Confidential

Path converters

URL 변수의 타입을 지정
(`str`, `int` 등 5가지 타입 지원)

https://docs.djangoproject.com/en/4.2/topics/http/urls/#path-converters

Confidential

Variable routing 실습 (1/2)

```
# urls.py

urlpatterns = [
    path('hello/<str:name>/', views.greeting),
]
```

```
# views.py

def greeting(request, name):
    context = {
        'name': name,
    }
return render(request, 'articles/greeting.html', context)
```

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

{% extends 'base.html' %}

{% block content %}
  <h1>Greeting</h1>
  <h3>{{ name }}님 안녕하세요 !</h3>
{% endblock content %}
```

Confidential

Variable routing 실습 (2/2)

```
# urls.py

urlpatterns = [
    path('articles/<int:num>/', views.detail),
]
```

```
# views.py

def detail(request, num):
    context = {
        'num': num,
    }
return render(request, 'articles/detail.html', context)
```

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

{% extends 'base.html' %}

{% block content %}
  <h1>Detail</h1>
  <h3>{{ num }}번 글 입니다.</h3>
{% endblock content %}
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

App과 URL

Confidential

App URL mapping

각 앱에 URL을 정의하는 것

- 프로젝트와 각 앱이 URL을 나누어 관리를 편하게 하기 위함

Confidential

2번째 앱 pages 생성 후 발생할 수 있는 문제

- view 함수 이름이 같거나 같은 패턴의 URL 주소를 사용하게 되는 경우
- 아래 코드와 같이 해결해 볼 수 있으나 더 좋은 방법이 필요

➢ "URL을 각자 app에서 관리하자"

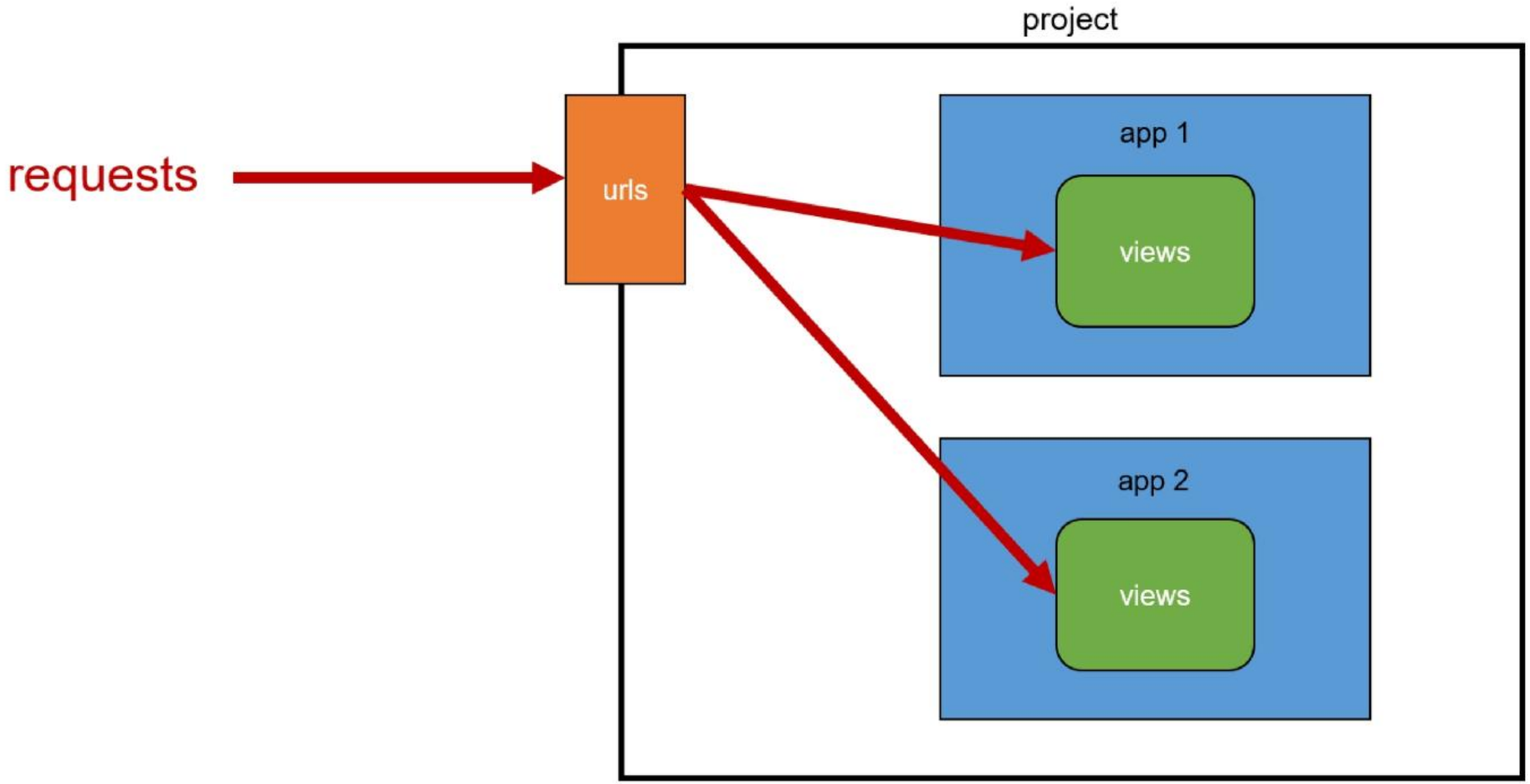
```
# firstpjt/urls.py

from articles import views as articles_views
from pages import views as pages_views


urlpatterns = [
    …,
    path('pages', pages_views.index),
]
```

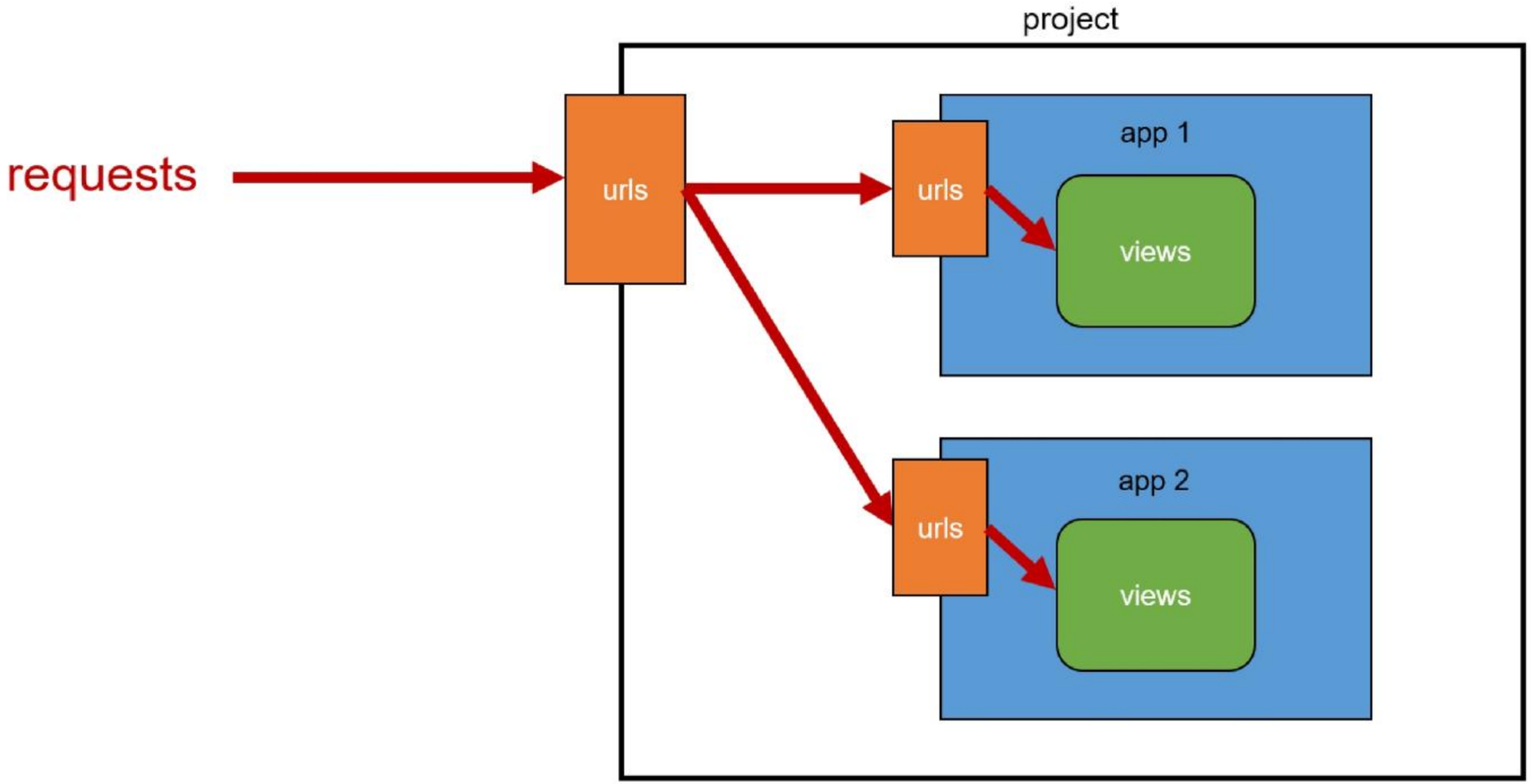
Confidential

기존 url 구조



Confidential

변경된 url 구조



Confidential

url 구조 변화

```python
# firstpjt/urls.py

from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    # path('index/', views.index),
    # path('dinner/', views.dinner),
    # path('search/', views.search),
    # path('throw/', views.throw),
    # path('catch/', views.catch),
    # path('articles/<int:num>/', views.detail),
    # path('hello/<str:name>/', views.greeting),
    # path('index/', views.index),

    path('articles/', include('articles.urls')),
    path('pages/', include('pages.urls')),
]
```

```python
# articles/urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('index/', views.index),
    path('dinner/', views.dinner),
    path('search/', views.search),
    path('throw/', views.throw),
    path('catch/', views.catch),
    path('<int:num>/', views.detail),
    path('hello/<str:name>/', views.greeting),
]
```

```python
# pages/urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('index/', views.index),
]
```

Confidential

include()

프로젝트 내부 앱들의 URL을 참조할 수 있도록 매핑하는 함수

- URL의 일치하는 부분까지 잘라내고,
남은 문자열 부분은 후속 처리를 위해 include된 URL로 전달

Confidential

include 적용

변경된 프로젝트의 urls.py

```
# firstpjt/urls.py

from django.urls import path, include

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

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

URL 이름 지정

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Naming URL patterns

Confidential

url 구조 변경에 따른 문제점

- 기존 ‘articles/’ 주소가 ‘articles/index/’로 변경됨에 따라 해당 url을 사용하는 모든 위치를 찾아가 변경해야 함

➢ “URL에 이름을 지어주면 이름만 기억하면 되지 않을까?”

```
# firstpjt/urls.py

path('articles/', include('articles.urls'))
```

```
# articles/urls.py

path('index/', views.index, name='index')
```

Confidential

Naming URL patterns

URL에 이름을 지정하는 것
(path 함수의 name 인자를 정의해서 사용)

Confidential

Naming URL patterns 적용

path 함수의 name 키워드 인자를 정의

```
# articles/urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('index/', views.index, name='index’),
    path('dinner/', views.dinner, name='dinner’),
    path('search/', views.search, name='search’),
    path('throw/', views.throw, name='throw’),
    path('catch/', views.catch, name='catch’),
    path('articles/<int:num>/', views.detail, name='detail’),
    path('hello/<str:name>/', views.greeting, name='greeting'),
]
```

```
# pages/urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('index/', views.index, name='index’),
]
```

Confidential

URL 표기 변화

url을 작성하는 모든 곳에서 변경

(a 태그의 href 속성 값 뿐만 아니라 form의 action 속성 등도 포함)

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

{% extends 'base.html' %}

{% block content %}
  <h1>Hello, {{ name }}</h1>
  <a href="/dinner/">dinner</a>
  <a href="/search/">search</a>
  <a href="/throw/">throw</a>
{% endblock content %}
```

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

{% extends 'base.html' %}

{% block content %}
  <h1>Hello, {{ name }}</h1>
  <a href="{% url 'dinner' %}">dinner</a>
  <a href="{% url 'search' %}">search</a>
  <a href="{% url 'throw' %}">throw</a>
{% endblock content %}
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

DTL URL tag

Confidential

'url' tag

```
{% url 'url name' arg1 arg2 %}
```

주어진 URL 패턴의 이름과 일치하는 절대 경로 주소를 반환

Confidential

url tag 적용 후 브라우저 출력 확인

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

URL 이름 공간

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

app_name 속성

Confidential

URL 이름 지정 후 남은 문제

- articles 앱의 url 이름과 pages 앱의 url 이름이 같은 상황
- 단순히 이름만으로는 완벽하게 분리할 수 없음

➢ "이름에 성(key)을 붙이자"

```
# articles/urls.py

path('index/', views.index, name='index')
```

```
# pages/urls.py

path('index/', views.index, name='index')
```

Confidential

'app_name' 속성 지정

app_name 변수 값 설정

```
# articles/urls.py

app_name = 'articles'
urlpatterns = [
    ...,
]
```

```
# pages/urls.py

app_name = 'pages'
urlpatterns = [
    ...,
]
```

Confidential

URL tag의 최종 변화

마지막으로 url 태그가 사용하는 모든 곳의 표기 변경하기

```
{% url 'app_name:path_name' %}
```

```
{% url 'index' %}
```

→

```
{% url 'articles:index' %}
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

참고

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

추가 템플릿 경로

Confidential

추가 템플릿 경로 지정 (1/2)

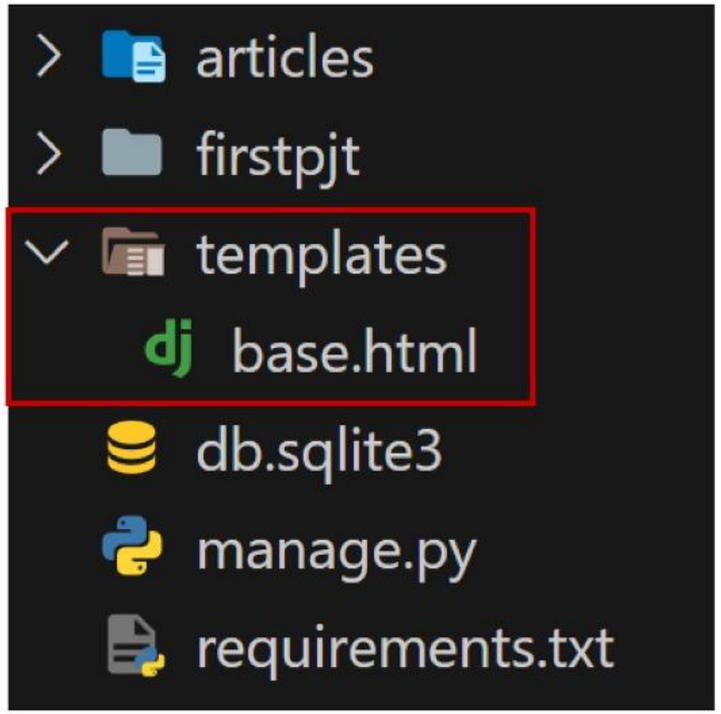
템플릿 기본 경로 외 커스텀 경로 추가하기

```
# settings.py

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            BASE_DIR / 'templates',
        ],
        'APP_DIRS': True,
        'OPTIONS': {
            …
        },
    },
]
```

Confidential

추가 템플릿 경로 지정 (2/2)



새로운 템플릿 경로 생성

```
{% extends 'base.html' %}
```

각 하위 템플릿에서
extends 경로가 수정됨

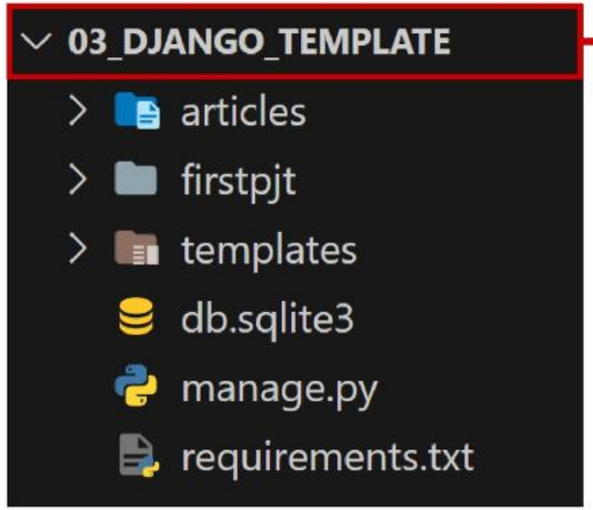
Confidential

BASE_DIR

settings에서 경로지정을 편하게 하기 위해 최상단 지점을 지정해 둔 변수

```
# settings.py

BASE_DIR = Path(__file__).resolve().parent.parent
```



BASE_DIR 경로 위치

- Python의 객체 지향 파일 시스템 경로
- https://docs.python.org/ko/3.9/library/pathlib.html#module-pathlib

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

DTL 주의사항

Confidential

DTL 주의사항

- Python처럼 일부 프로그래밍 구조(if, for 등)를 사용할 수 있지만 명칭을 그렇게 설계 했을 뿐이지 Python 코드로 실행되는 것이 아니며 Python과는 관련 없음
- 프로그래밍적 로직이 아니라 표현을 위한 것임을 명심하기
- 프로그래밍적 로직은 되도록 view 함수에서 작성 및 처리할 것
- 공식문서를 참고해 다양한 태그와 필터 사용해보기
  - https://docs.djangoproject.com/en/4.2/ref/templates/builtins/

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Trailing Slashes

Confidential

URL의 Trailing Slashes

- Django는 URL 끝에 '/'가 없다면 자동으로 붙임
- "기술적인 측면에서, `foo.com/bar`와 `foo.com/bar/`는 서로 다른 URL"
  - 검색 엔진 로봇이나 웹 트래픽 분석 도구에서는 이 두 주소를 서로 다른 페이지로 보기 때문
- 그래서 Django는 검색 엔진이 혼동하지 않게 하기 위해 무조건 붙이는 것을 선택한 것
- 그러나 모든 프레임워크가 이렇게 동작하는 것은 아니니 주의

SAMSUNG
SW
ACADEMY
FOR
YOUTH

함께가요 미래로!
Enabling People

다음 시간에 만나요!

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Django 03
Model

SAMSUNG SW ACADEMY FOR YOUTH

• INDEX

- Model
  - model class
- Model Field
  - Field types
  - Field options
- Migrations
  - 추가 Migrations
- Admin site
- 참고
  - 데이터베이스 초기화
  - Migrations 관련
  - SQLite

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Model

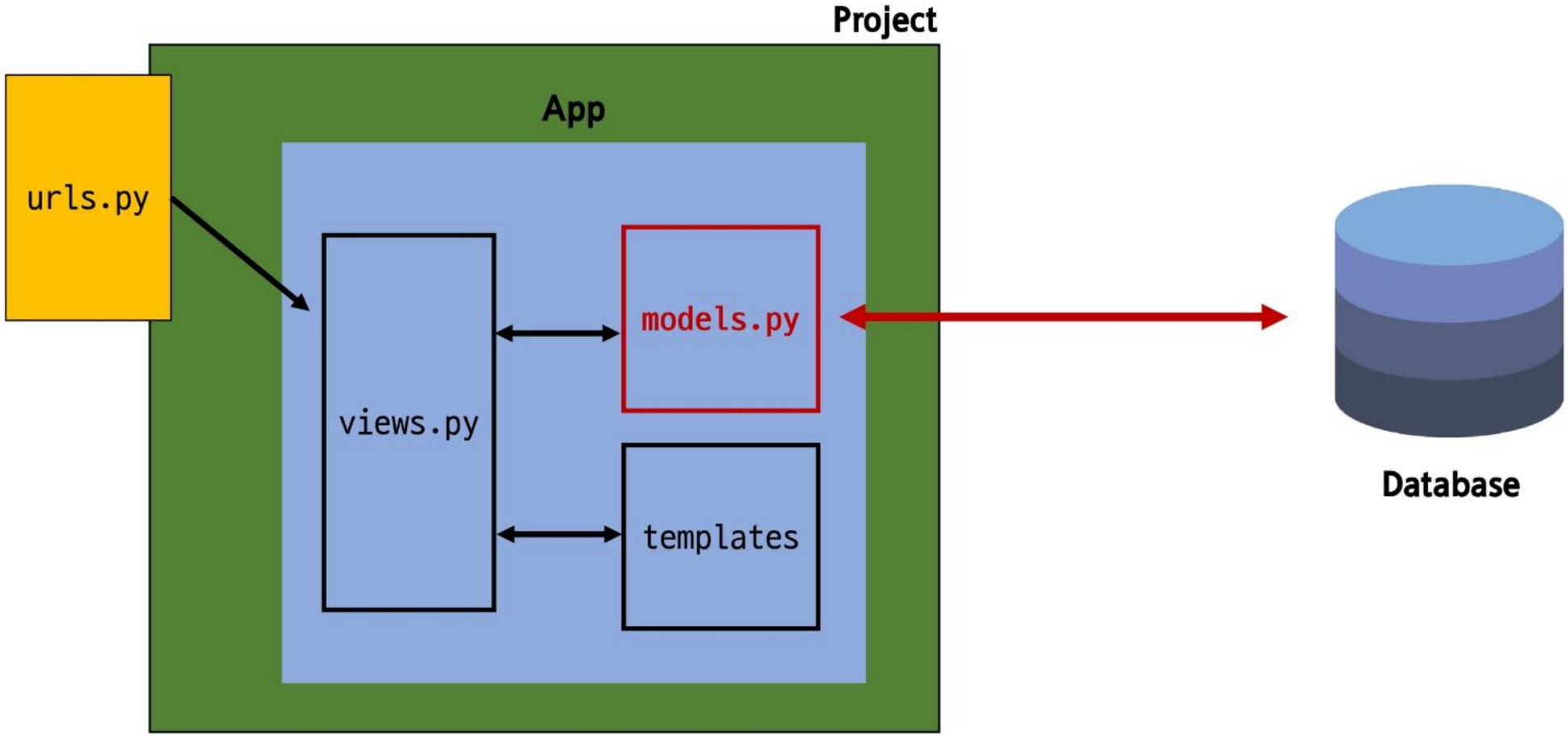
SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

model class

Confidential

Model을 통한 DB(데이터베이스) 관리



Confidential

Django Model

DB의 테이블을 정의하고 데이터를 조작할 수 있는 기능들을 제공

- 테이블 구조를 설계하는 '청사진(blueprint)'

Confidential

model 클래스 작성

```
# articles/models.py

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

Confidential

model 클래스 살펴보기 (1/4)

작성한 모델 클래스는 최종적으로 DB에 다음과 같은 테이블 구조를 만듦

```
# articles/models.py

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

❖ id 필드는 Django가 자동 생성

id	title	content
..	..	..
..	..	..

'모델 클래스 == 테이블 설계도'

Confidential

model 클래스 살펴보기 (2/4)

- `django.db.models` 모듈의 `Model`이라는 부모 클래스를 상속받음
- `Model`은 model에 관련된 모든 코드가 이미 작성 되어있는 클래스
  - https://github.com/django/django/blob/main/django/db/models/base.py#L460

➢ 개발자는 가장 중요한 테이블 구조를 어떻게 설계할지에 대한 코드만 작성하도록 하기 위한 것 (상속을 활용한 프레임워크의 기능 제공)

```
# articles/models.py

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

Confidential

model 클래스 살펴보기 (3/4)

- 클래스 변수명
  - 테이블의 각 "필드(열) 이름"

```
# articles/models.py

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

id	title	content
..	..	..
..	..	..

Confidential

model 클래스 살펴보기 (4/4)

- Model Field
  - 데이터베이스 테이블의 열(column)을 나타내는 중요한 구성 요소
  - "데이터의 유형"과 "제약 조건"을 정의

```
# articles/models.py

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Model Field

Confidential

Model Field

DB 테이블의 필드(열)을 정의하며,
해당 필드에 저장되는 데이터 타입과 제약조건을 정의

Field types Field options

https://docs.djangoproject.com/en/4.2/ref/models/fields/

Confidential

Model Field 구성

1. Field types (필드 유형)
   - 데이터베이스에 저장될 “데이터의 종류”를 정의

2. Field options (필드 옵션)
   - 필드의 “동작”과 “제약 조건”을 정의

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Field types

Confidential

Field Types

데이터베이스에 저장될 "데이터의 종류"를 정의
(models 모듈의 클래스로 정의되어 있음)

```
class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

Confidential

주요 필드 유형

- 문자열 필드
  - CharField, TextField
- 숫자 필드
  - IntegerField, FloatField
- 날짜/시간 필드
  - DateField, TimeField, DateTimeField
- 파일 관련 필드
  - FileField, ImageField

Confidential

CharField()

제한된 길이의 문자열을 저장
(필드의 최대 길이를 결정하는 max_length는 필수 옵션)

Confidential

TextField()

길이 제한이 없는 대용량 텍스트를 저장
(무한대는 아니며 사용하는 시스템에 따라 달라짐)

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로! Enabling People

Field options

Confidential

Field Options

필드의 "동작"과 "제약 조건"을 정의

```
class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

Confidential

주요 필드 옵션

- `null`
  - 데이터베이스에서 NULL 값을 허용할지 여부를 결정 (기본값: False)
- `blank`
  - form에서 빈 값을 허용할지 여부를 결정 (기본값: False)
- `default`
  - 필드의 기본값을 설정

Confidential

제약 조건

Constraint

특정 규칙을 강제하기 위해 테이블의 열이나 행에
적용되는 규칙이나 제한사항

- ex) 숫자만 저장되도록, 문자가 100자 까지만 저장되도록 하는 등

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Migrations

Confidential

Migrations

---

model 클래스의 변경사항(필드 생성, 수정 삭제 등)을
DB에 최종 반영하는 방법

Confidential

Migrations 과정

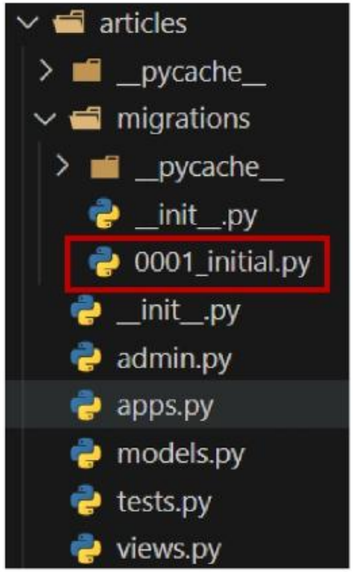
```
# articles/models.py

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
```

model class

(설계도 초안)

makemigrations →



migration 파일

(최종 설계도)

migrate →

db.sqlite3

(DB)

Confidential

Migrations 핵심 명령어 2가지

```
$ python manage.py makemigrations
```

model class를 기반으로
최종 설계도(migration) 작성

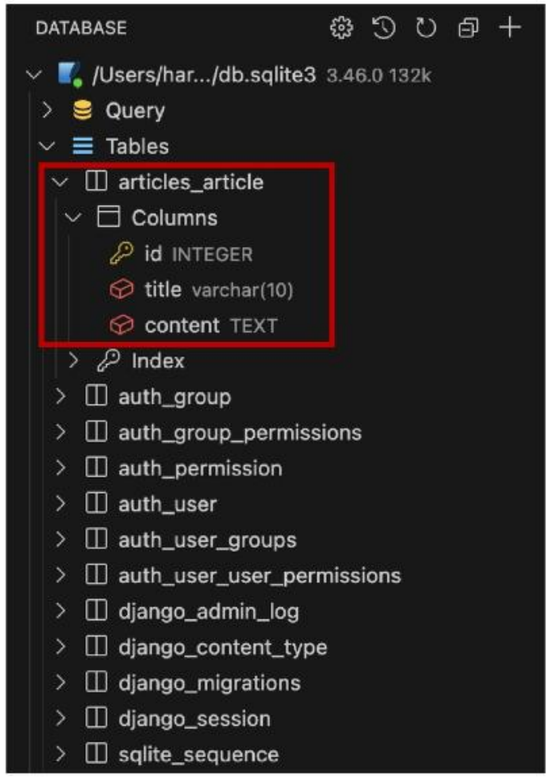
```
$ python manage.py migrate
```

최종 설계도를
DB에 전달하여 반영

Confidential

migrate 후 DB 내에 생성 된 테이블 확인

Article 모델 클래스로 만들어진 `articles_article` 테이블



SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

추가 Migrations

Confidential

이미 생성된 테이블에 필드를 추가해야 한다면?

id	title	content	created_at	updated_at
..	..	..	..	..
..	..	..	..	..
..	..	..	..	..

Confidential

추가 모델 필드 작성 (1/5)

```
# articles/models.py

class Article(models.Model):
    title = models.CharField(max_length=10)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
```

Confidential

DateTimeField의 필드 옵션 (optional)

auto_now

데이터가 저장될 때마다
자동으로 현재 날짜시간을 저장

auto_now_add

데이터가 처음 생성될 때만
자동으로 현재 날짜시간을 저장

Confidential

추가 모델 필드 작성 (2/5)

- 이미 기존 테이블이 존재하기 때문에 필드를 추가 할 때 필드의 기본 값 설정이 필요
- 1번은 현재 대화를 유지하면서 직접 기본 값을 입력 하는 방법
- 2번은 현재 대화에서 나간 후 models.py에 기본 값 관련 설정을 하는 방법

```
$ python manage.py makemigrations
```

```
It is impossible to add the field 'created_at' with 'auto_now_add=True' to article
without providing a default. This is because the database needs something to populate
existing rows.
 1) Provide a one-off default now which will be set on all existing rows
 2) Quit and manually define a default value in models.py.
Select an option:
```

Confidential

추가 모델 필드 작성 (3/5)

- 추가하는 필드의 기본 값을 입력해야 하는 상황
- 날짜 데이터이기 때문에 직접 입력하기 보다 Django가 제안하는 기본 값을 사용하는 것을 권장
- 아무것도 입력하지 않고 enter를 누르면 Django가 제안하는 기본 값으로 설정 됨

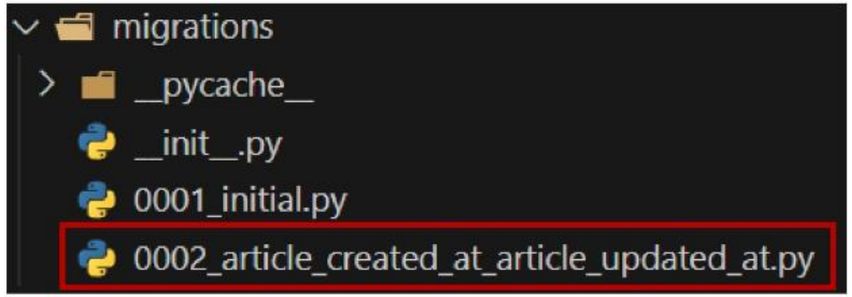
```
Please enter the default value as valid Python.
Accept the default 'timezone.now' by pressing 'Enter' or provide another value.
The datetime and django.utils.timezone modules are available, so it is possible to
provide e.g. timezone.now as a value.
Type 'exit' to exit this prompt
[default: timezone.now] >>>
```

Confidential

추가 모델 필드 작성 (4/5)

- migrations 과정 종료 후 2번째 migration 파일이 생성됨을 확인
- 이처럼 Django는 설계도를 쌓아가면서 추후 문제가 생겼을 시 복구하거나 되돌릴 수 있도록 함 (마치 '`git commit`'과 유사)

```
Migrations for 'articles':
  articles\migrations\0002_article_created_at_article_updated_at.py
    - Add field created_at to article
    - Add field updated_at to article
```

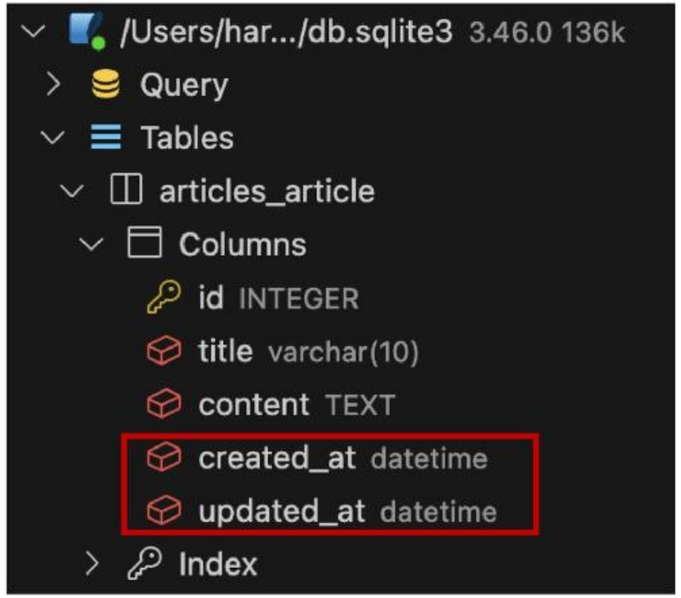


Confidential

추가 모델 필드 작성 (5/5)

migrate 후 테이블 필드 변화 확인

```
$ python manage.py migrate
```



Confidential

model class에 변경사항(1)이 생겼다면,
반드시 새로운 설계도를 생성(2)하고,
이를 DB에 반영(3)해야 한다.

1. `model class` 변경 → 2. `makemigrations` → 3. `migrate`

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Admin site

Confidential

Automatic admin interface

Django가 추가 설치 및 설정 없이 자동으로 제공하는
관리자 인터페이스

- 데이터 확인 및 테스트 등을 진행하는데 매우 유용

Confidential

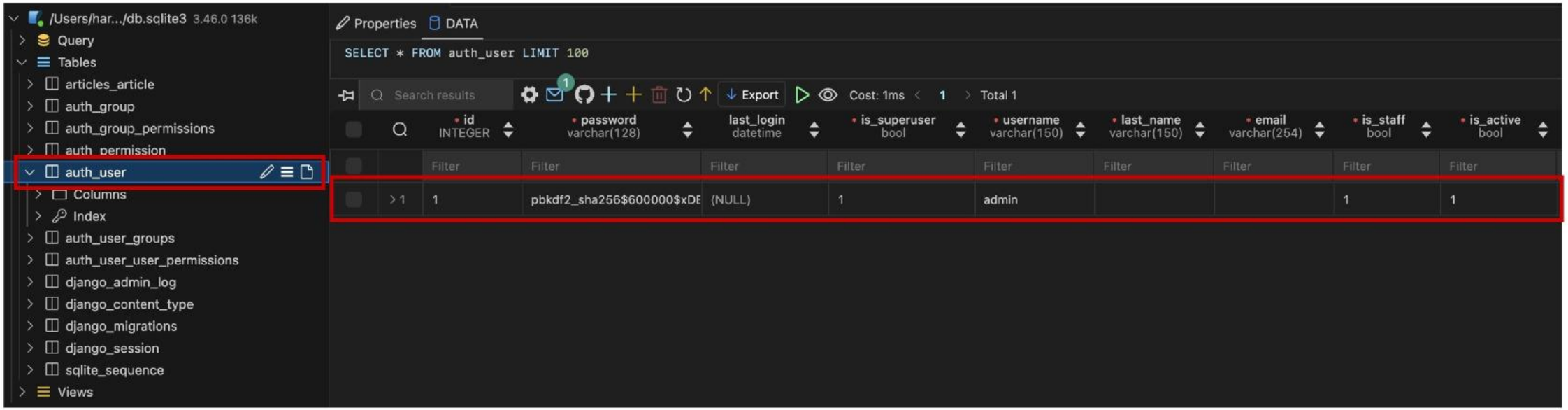
1. admin 계정 생성

- email은 선택사항이기 때문에 입력하지 않고 진행 가능
- 비밀번호 입력 시 보안상 터미널에 출력되지 않으니 무시하고 입력 이어가기

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

Confidential

2. DB에 생성된 admin 계정 확인



Confidential

3. admin에 모델 클래스 등록

admin.py에 작성한 모델 클래스를 등록해야만 admin site에서 확인 가능

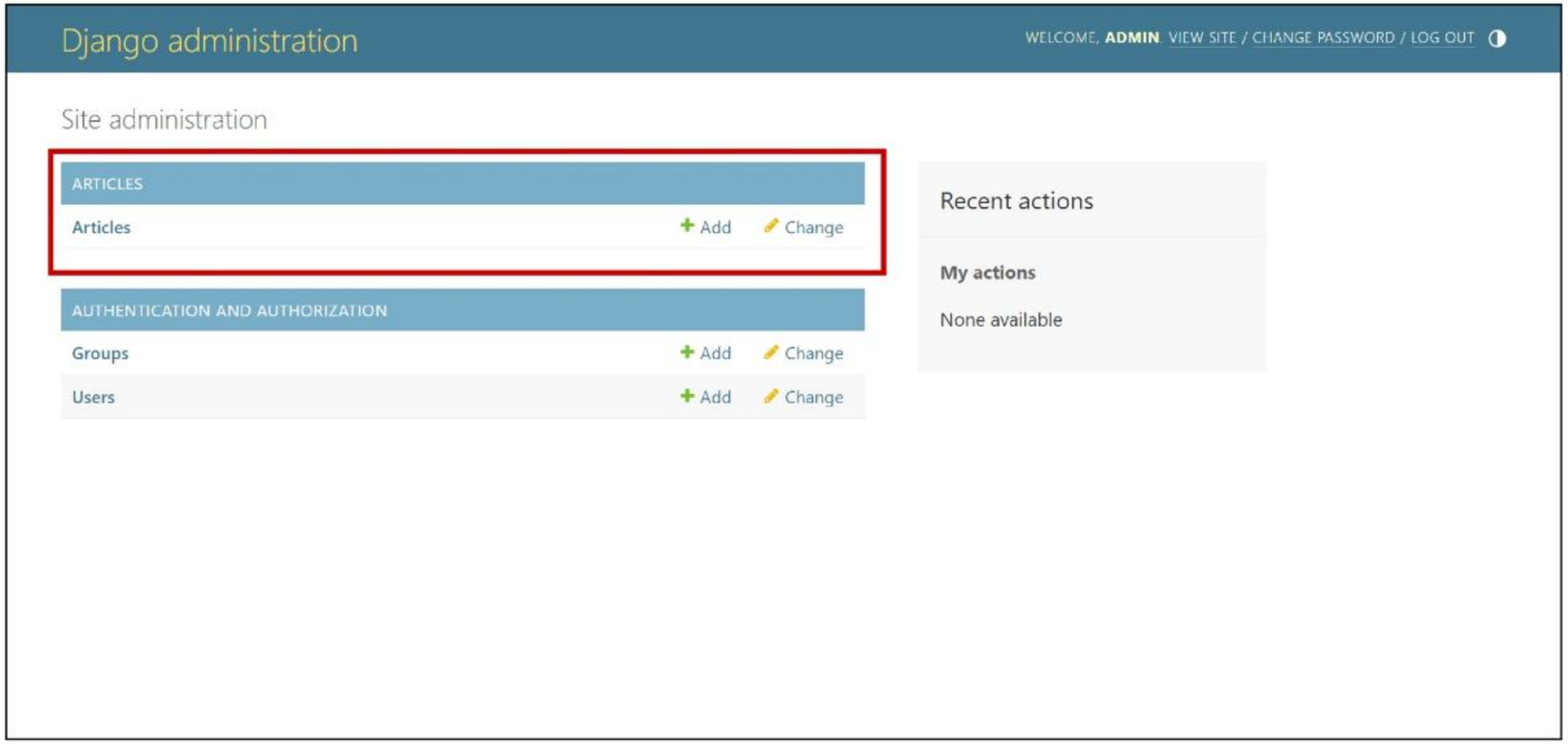
```
# articles/admin.py

from django.contrib import admin
from .models import Article


admin.site.register(Article)
```

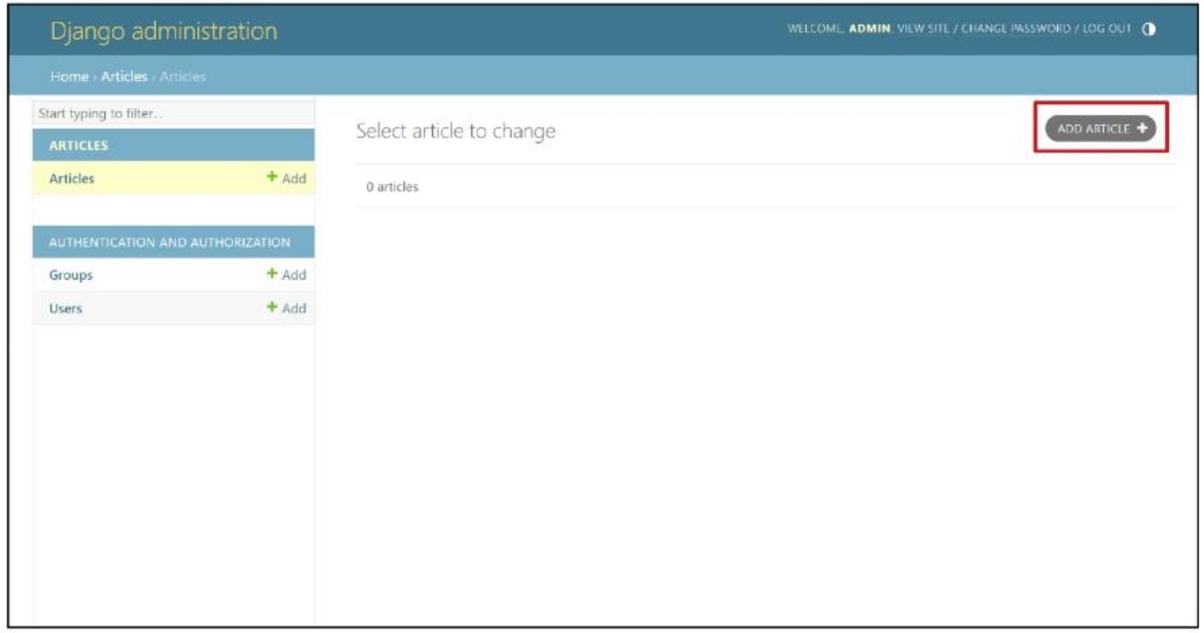
Confidential

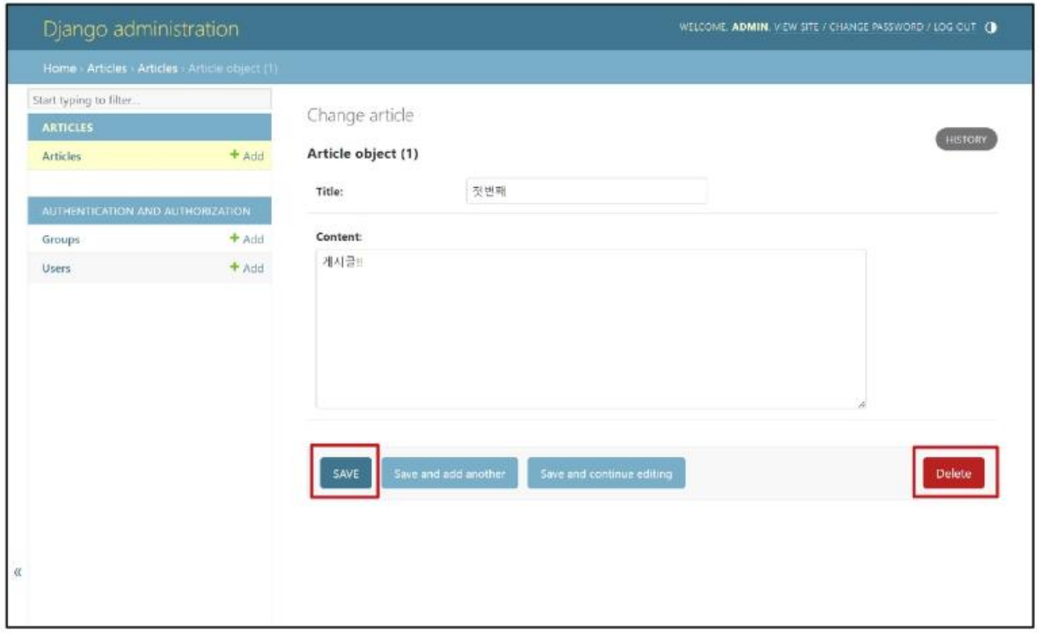
4. admin site 로그인 후 등록된 모델 클래스 확인



Confidential

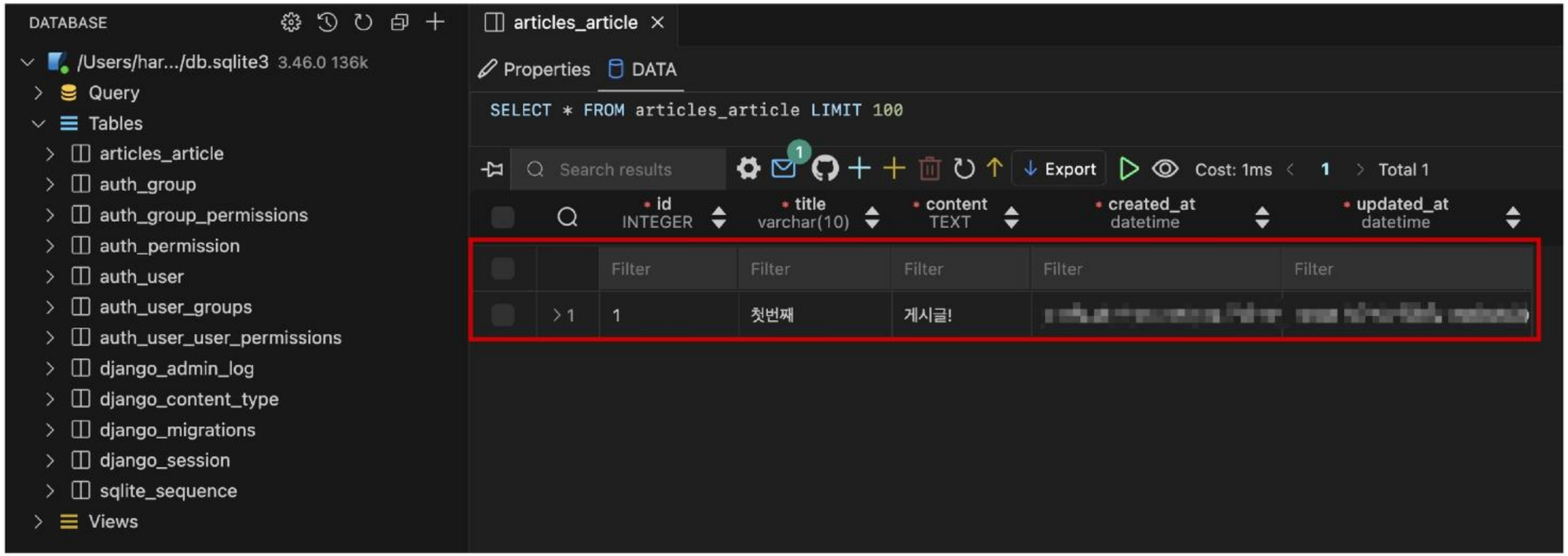
5. 데이터 생성, 수정, 삭제 테스트





Confidential

6. 테이블 확인



SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

이어서..

삼성 청년 SW 아카데미

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

참고

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

데이터베이스 초기화

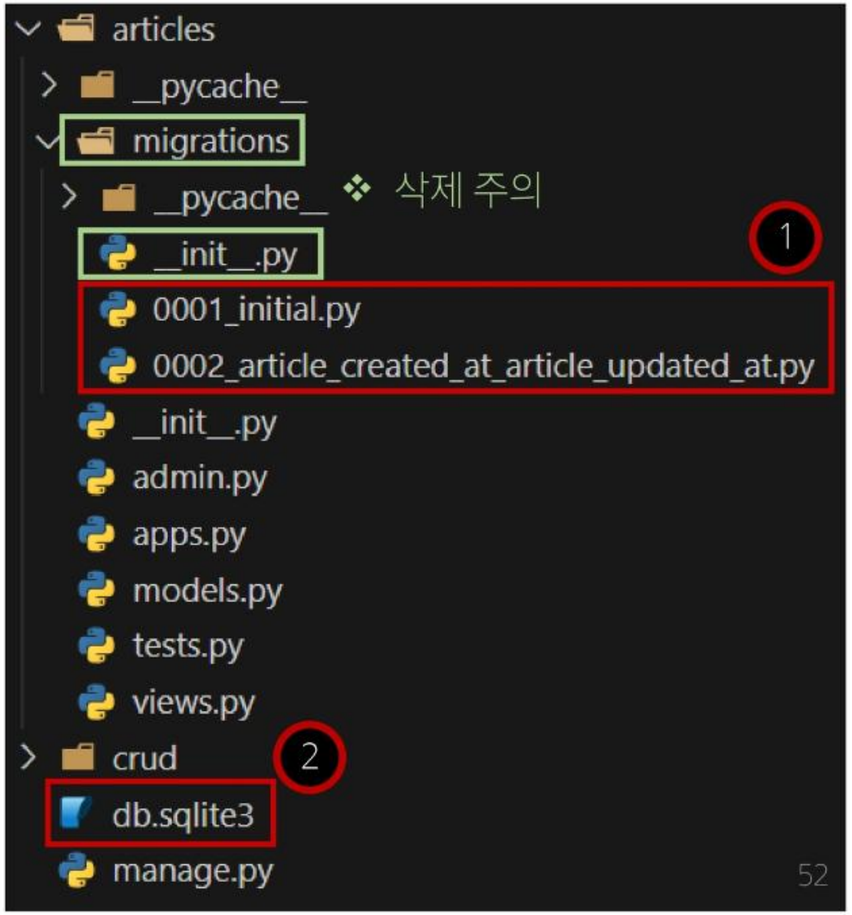
Confidential

데이터베이스 초기화

1. migration 파일 삭제
2. db.sqlite3 파일 삭제

❖ 아래 파일과 폴더를 지우지 않도록 주의

- `__init__.py`
- `migrations` 폴더



SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Migrations 관련

Confidential

Migrations 기타 명령어

```
$ python manage.py showmigrations
```

- migrations 파일들이 migrate 됐는지 안됐는지 여부를 확인하는 명령어
- [X] 표시가 있으면 migrate가 완료되었음을 의미

```
$ python manage.py sqlmigrate articles 0001
```

- 해당 migrations 파일이 SQL 언어(DB에서 사용하는 언어)로 어떻게 번역 되어 DB에 전달되는지 확인하는 명령어

Confidential

첫 migrate 시 출력 내용이 많은 이유는?

Django 프로젝트가 동작하기 위해 미리 작성 되어있는
기본 내장 app들에 대한 migration 파일들이 함께 migrate 되기 때문

```
$ 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
```

```python
# settings.py

INSTALLED_APPS = [
    'articles',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

SQLite

Confidential

SQLite

데이터베이스 관리 시스템 중 하나이며
Django의 기본 데이터베이스로 사용됨
(파일로 존재하며 가볍고 호환성이 좋음)

SAMSUNG
SW
ACADEMY
FOR
YOUTH

함께가요 미래로!
Enabling People

다음 시간에 만나요!

삼성 청년 SW 아카데미