



Session 9

# Authentication

NEXT[9].강단비



# 들어가며



(7기, 권XX, 천명)

## (나름 어문학도의 잡소리)

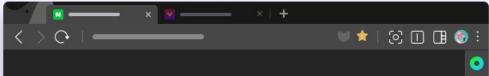
- 여러분이 배우는 건 파이썬이란 언어잖아요? 자 영어 처음배울 때 생각 해보세요(살다오신분 제외 부럽다) 단어도 처음 배우면 외워야할 것도 짱 많고 문법도 어렵고 학교에서 스펠링 시험 보면 뭐 여러분들이야 고대 생이시니까 100점만 맞으셔왔겠지만 틀리기 마련이잖아요?? 저희가 컴퓨터입니까?? 0과 1밖에 모르는 사람들이 아니잖아요?? 하루에 100시간 이상 코딩에 투자했습니까?? (이상 문과) 그것도 아니잖아요?? 그러니까 틀리는건 너무나 자연스럽고 당연한 과정이라는걸 꼭 아셔주시고 좌절하지 말고!!!! 시간투자하셔서!!!! 교안대로 따라치기만 하셔도!!!! 코딩엔!!!! 익숙해지실수있다!!!!!!! 라는 말을!!!! 하고 싶었어요!!!!!!! 으아!!!!



# Authentication 이란?

## Authentication

인증(authentication)은  
참이라는 근거가 있는 무언가를 확인하거나 확증하는 행위이다.  
**사람을 인증하는 것은 사람들의 신분을 구성하는 것을 말한다.**



여러분의 눈은 소중하니까, 지금 바로 다크 모드를 켜세요!

다운로드

3일 동안 보지 않기 X

네이버를 시작페이지로 &gt; | 쿠팡네이버 해피빈



NAVER

▼
Q


메일 카페 블로그 지식iN 쇼핑 쇼핑 LIVE Pay ▶TV 사전 뉴스 증권 부동산 지도 VIBE 책 웹툰 더보기 ▾

미세 ☺ 좋음 | 초미세 😞 보통 장위동



연합뉴스 &gt; 이성윤, 사실상 '버티기' 모드...거취 논란 커질 듯

네이버뉴스 · 연예 스포츠 경제

NEWS N 뉴스스탠드 &gt; 구독한 언론사 · 전체언론사



The Korea Herald	미디어 신문	ZDNet Korea	YTN	SBS	서울경제
OSN	일간스포츠	연합뉴스TV	아시아경제	OhmyNews	한국경제TV
스포츠서울	MBC	세계일보	맥스무비	대한경제	비즈한국

네이버를 더 안전하고 편리하게 이용하세요

NAVER 로그인

아이디 · 비밀번호찾기

회원가입

환율 | USD 1,125.50 ▲ 6.50 +0.58%

NAVER

흔들어 바로  
QR체크인

더 간편해진 네이버앱 &gt;

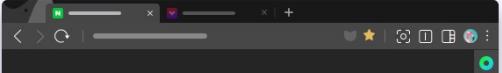


다크 모드로 보기

N 트렌드쇼핑 &gt;

상품 쇼핑몰 MEN

↑



여러분의 눈은 소중하니까, 지금 바로 다크 모드를 켜세요!

다운로드

3일 동안 보지 않기 X

네이버를 시작페이지로 &gt; | 쥬니어네이버 해피빈



NAVER

 ▶ Q[메일](#) [카페](#) [블로그](#) [지식iN](#) [쇼핑](#) [LIVE](#) [Pay](#) [▶TV](#) [사전](#) [뉴스](#) [증권](#) [부동산](#) [지도](#) [VIBE](#) [책](#) [웹툰](#) [더보기 ▾](#)

17.0° 맑음 14.0° / 29.0° 장위동

[연합뉴스](#) > 민주, '임박노' 중 1+α 지명철회 요구...青 수용 기류

네이버뉴스 · 연예 스포츠 경제

[뉴스스탠드](#) > 구독한 언론사 · 전체언론사

헤럴드경제

헤럴드경제 2021.05.13. 00:37 편집



"손정민씨 친구, 실종 당일 오전 2시18분께 휴대전화 사용"  
인천 노래방 피살손님 훼손된 시신 발견...업주, 범행 자백  
진성준 "임·박·노 낙마 요구, 의회 민주주의와 맞지 않아" 공...  
화이자 백신 맞고 2분 만에 의식 잃은 80대 여 사망  
"30억원 벌고 퇴사?" 도지코인 대박에 사표 던지는 직장인들  
"수천만원 보너스 잔치" 만년 적자 중소게임사의 '반전'

토재이님 | 내정보 🔒  
tojay24@naver.com  
메일 999+ 쪽지 51

1 알림 N MY구독 999+ 메일 50 카페 블로그 >

증시 | 나스닥 13,215.49 ▼ 297.35 -2.22%

AD

이제 간편하게 동네시장 장보기

시장표 반반족발도  
2시간 안에 배달됩니다  
지금 구매하기 >

N 트렌드쇼핑 &gt;

상품 쇼핑몰 MEN

다크 모드로 보기

↑



## 오늘 할 일

- Issue 1** 회원가입 / 로그인 / 로그아웃 기능 구현
- Issue 2** 로그인 상태에 따라 다른 navbar
- Issue 3** 글의 작성자만 Post, Comment 수정, 삭제 권한 부여
- Issue 4** 카카오 API를 이용한 Social Login



# 세션 준비

복불 합시다. 갈 길이 매우매우 멈춥니다…

```
$ git clone https://github.com/domino04/session-auth.git  
$ cd session-auth  
$ rm Pipfile  
$ rm Pipfile.lock  
$ pipenv shell  
$ pipenv install django  
$ cd crudproject  
$ python manage.py migrate  
$ python manage.py createsuperuser
```



# Model 만들기

User 모델은 이미 존재하므로,  
Model을 따로 만들 필요 없고 import만 해주면 된다.

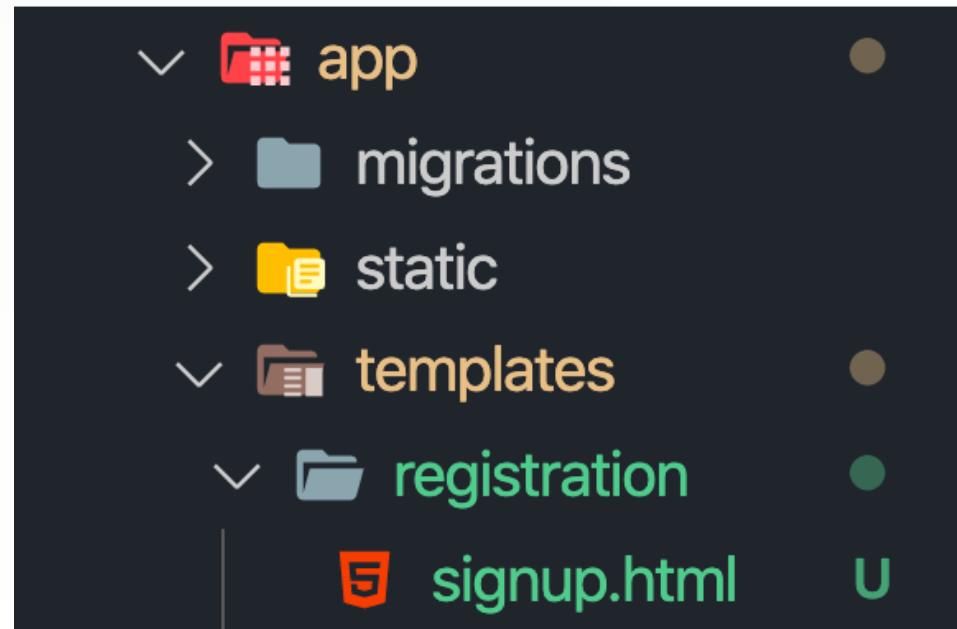
Crudproject > app > views.py

```
from django.shortcuts import render, redirect  
from .models import Post, Comment  
from django.contrib.auth.models import User
```



# Template 만들기

Templates 폴더에 registration 폴더를 만든다.  
signup.html 파일을 만든다.





# Template 만들기 (수정)

Crudproject > app > templates > registration > signup.html

```
{% extends 'base.html' %} (추가)  
{% block content %}  
    <h1> 회원가입 </h1>  
    <form method="POST">  
        {% csrf_token %}  
        <input type="text" name="username" placeholder="이름">  
        <input type="password" name="password" placeholder="비밀번호">  
        <button type="submit">가입하기</button>  
    </form>  
{% endblock content %}
```



# View 만들기

Crudproject > app > views.py

```
def signup(request):
    if (request.method == 'POST'):
        new_user = User.objects.create_user(
            username = request.POST['username'],
            password = request.POST['password']
        )
        return redirect('home')

    return render(request, 'registration/signup.html')
```



# Url 설정하기

crudproject > crudproject > urls.py

```
urlpatterns = [
    path('registration/signup', views.signup, name="signup"),
    path('admin/', admin.site.urls),
    path('', views.home, name="home"),
    path('new/', views.new, name="new"),
    path('detail/<int:post_pk>', views.detail, name="detail"),
    path('edit/<int:post_pk>', views.edit, name="edit"),
    path('delete/<int:post_pk>', views.delete, name="delete"),
    path('delete_comment/<int:post_pk>/<int:comment_pk>',
         views.delete_comment, name="delete_comment")
]
```



# base.html 수정

Auth > app > templates > base.html

```
<body>
    <div class="nav-bar">
        <a class="nav-item" href="{% url 'home' %}>HOME</a>
        <a class="nav-item">로그인</a>
        <a class="nav-item" href="{% url 'signup' %}>회원가입</a>
        <a class="nav-item">로그아웃</a>
    </div>
    {% block content %}
    {% endblock content %}
</body>
```



# 회원가입 시도

localhost:8000/admin

Site administration

APP

Comments	<a href="#"> Add</a>	<a href="#"> Change</a>
Posts	<a href="#"> Add</a>	<a href="#"> Change</a>

AUTHENTICATION AND AUTHORIZATION

Groups	<a href="#"> Add</a>	<a href="#"> Change</a>
Users	<a href="#"> Add</a>	<a href="#"> Change</a>



# 회원가입 시도

localhost:8000/admin

Select user to change

Search

Action: -----  0 of 2 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	admin				<input checked="" type="checkbox"/>
<input type="checkbox"/>	lee				<input type="checkbox"/>

2 users



# Error!

같은 **username**으로 회원가입을 시도하면  
**오류**가 난다

## IntegrityError at /auth/signup

UNIQUE constraint failed: auth\_user.username

Request Method: POST

Request URL: http://localhost:8000/auth/signup

Django Version: 3.0.6

Exception Type: IntegrityError

Exception Value: UNIQUE constraint failed: auth\_user.username

Exception Location: /Users/kyoungyeon/.local/share/virtualenvs/crud-1-n-GAwKw4y0/lib/python3.7/site-packages/django/db/backends/sqlite3/base.py in execute, line 396

Python Executable: /Users/kyoungyeon/.local/share/virtualenvs/crud-1-n-GAwKw4y0/bin/python

Python Version: 3.7.6

Python Path: [ '/Users/kyoungyeon/Desktop/test/crud-1-n/crudproject',  
              '/usr/local/var/pyenv/versions/3.7.6/lib/python37.zip',  
              '/usr/local/var/pyenv/versions/3.7.6/lib/python3.7',  
              '/usr/local/var/pyenv/versions/3.7.6/lib/python3.7/lib-dynload',  
              '/Users/kyoungyeon/.local/share/virtualenvs/crud-1-n-GAwKw4y0/lib/python3.7/site-packages' ]

Server time: Sun, 24 May 2020 08:49:07 +0000



# 예외처리

Auth > app > view.py

```
def signup(request):
    if (request.method == 'POST'):
        found_user = User.objects.filter(username=request.POST['username'])
        if (len(found_user) > 0):
            error = 'username이 이미 존재합니다'
            return render(request, 'registration/signup.html', {
                'error' : error
            })
        new_user = User.objects.create(
            username = request.POST['username'],
            password = request.POST['password']
        )
        return redirect('home')

    return render(request, 'registration/signup.html')
```



# 예외처리

Auth > app > templates > registration > signup.html

```
{% block content %}  
    <h1> 회원가입 </h1>  
    {% if error %}  
        <h3 style="color: red;">{{ error }}</h3>  
    {% endif %}  
    <form method="POST">  
        {% csrf_token %}  
        <input type="" name="username" placeholder="이름">  
        <input type="password" name="password" placeholder="비밀번호">  
        <button type="submit">가입하기</button>  
    </form>  
{% endblock content %}
```



# nav-bar 수정하기

Auth > app > templates > base.html

```
<div class="nav-bar">
    <a class="nav-item" href="{% url 'home' %}">HOME</a>
    {% if user.is_authenticated %}
        <a class="nav-item">안녕하세요, {{ user.username }}님</a>
        <a class="nav-item">로그아웃</a>
    {% else %}
        <a class="nav-item">로그인</a>
        <a class="nav-item" href="{% url 'signup' %}">회원가입</a>
    {% endif %}
</div>
{% block content %}
{% endblock content %}
```



# 회원가입과 동시에 로그인 하기

모듈 import

Auth > app > view.py

```
from django.contrib import auth
```



# 회원가입과 동시에 로그인 하기

Auth > app > view.py

```
def signup(request):
    if (request.method == 'POST'):
        found_user = User.objects.filter(username=request.POST['username'])
        if (len(found_user) > 0):
            error = 'username이 이미 존재합니다'
            return render(request, 'registration/signup.html', {
                'error' : error
            })
        new_user = User.objects.create(
            username = request.POST['username'],
            password = request.POST['password']
        )
        auth.login(request, new_user)
        return redirect('home')

    return render(request, 'registration/signup.html')
```



# 로그인 만들기

Auth > app > templates > registration > login.html

```
{% block content %}  
    <h1> 로그인 </h1>  
    {% if error %}  
        <h3 style="color: red;">{{ error }}</h3>  
    {% endif %}  
    <form method="POST">  
        {% csrf_token %}  
        <input type="text" name="username" placeholder="아이디">  
        <input type="password" name="password" placeholder="비밀번호">  
        <button type="submit">로그인</button>  
    </form>  
{% endblock content %}
```



# 로그인 만들기

Auth > app > view.py

```
def login(request):
    if (request.method == 'POST'):
        found_user = auth.authenticate(
            username=request.POST['username'],
            password=request.POST['password']
        )

        if (found_user is None):
            error = '아이디 또는 비밀번호가 틀렸습니다'
            return render(request, 'registration/login.html', {
                'error' : error
            })
        auth.login(request, found_user)
        return redirect('home')

    return render(request, 'registration/login.html')
```



# 로그인 만들기

Auth > Auth > urls.py

```
urlpatterns = [
    path('registration/signup', views.signup, name="signup"),
    path('registration/login', views.login, name="login"),
    path('admin/', admin.site.urls),
    path('', views.home, name="home"),
    path('new/', views.new, name="new"),
    path('detail/<int:post_pk>', views.detail, name="detail"),
    path('edit/<int:post_pk>', views.edit, name="edit"),
    path('delete/<int:post_pk>', views.delete, name="delete"),
    path('delete_comment/<int:post_pk>/<int:comment_pk>',
         views.delete_comment, name="delete_comment")
]
```



# 로그인 만들기

Auth > app > templates > base.html

```
<div class="nav-bar">
    <a class="nav-item" href="{% url 'home' %}">HOME</a>
    {% if user.is_authenticated %}
        <div class="nav-item">안녕하세요, {{ user.username }}님</div>
        <a class="nav-item">로그아웃</a>
    {% else %}
        <a class="nav-item" href="{% url 'login' %}">로그인</a>
        <a class="nav-item" href="{% url 'signup' %}">회원가입</a>
    {% endif %}
</div>
{% block content %}
{% endblock content %}
```



## 로그아웃 만들기

Auth > app > view.py

```
def logout(request):
    auth.logout(request)

    return redirect('home.html')
```



# 로그아웃 만들기

Auth > Auth > urls.py

```
urlpatterns = [
    path('registration/signup', views.signup, name="signup"),
    path('registration/login', views.login, name="login"),
    path('registration/logout', views.logout, name="logout"),
    path('admin/', admin.site.urls),
    path('', views.home, name="home"),
    path('new/', views.new, name="new"),
    path('detail/<int:post_pk>', views.detail, name="detail"),
    path('edit/<int:post_pk>', views.edit, name="edit"),
    path('delete/<int:post_pk>', views.delete, name="delete"),
    path('delete_comment/<int:post_pk>/<int:comment_pk>',
         views.delete_comment, name="delete_comment")
]
```



# 로그아웃 만들기

Auth > app > templates > base.html

```
<div class="nav-bar">
    <a class="nav-item" href="{% url 'home' %}">HOME</a>
    {% if user.is_authenticated %}
        <div class="nav-item">안녕하세요, {{ user.username }}님</div>
        <a class="nav-item" href="{% url 'logout' %}">로그아웃</a>
    {% else %}
        <a class="nav-item" href="{% url 'login' %}">로그인</a>
        <a class="nav-item" href="{% url 'signup' %}">회원가입</a>
    {% endif %}
</div>
{% block content %}
{% endblock content %}
```



## 오늘 할 일

- Issue 1** ~~회원가입 / 로그인 / 로그아웃 기능 구현~~
- Issue 2** ~~로그인 상태에 따라 다른 navbar~~
- Issue 3** 글의 작성자만 Post, Comment 수정, 삭제 권한 부여
- Issue 4** 카카오 API를 이용한 Social Login



## 권한 부여하기

모듈 import

Auth > app > view.py

```
from django.contrib.auth.decorators import login_required
```



# 권한 부여

Auth > app > view.py

```
@login_required
def new(request):
    if (request.method == 'POST'):
        new_post = Post.objects.create(
            title = request.POST['title'],
            content = request.POST['content'],
        )
        return redirect('detail', new_post.pk)
    return render(request, 'new.html')
```



# 그렇다면...

????: 작성자를 구분하려면 어떻게 해야할까?

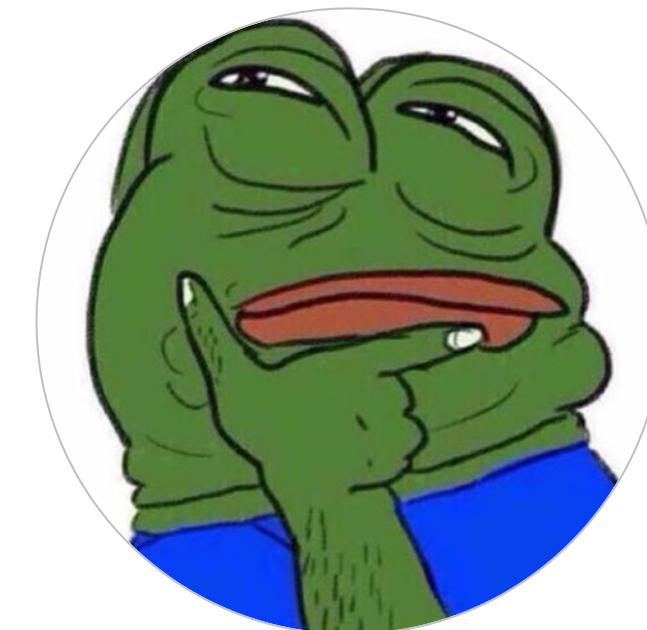
## 장고 모델 1:N

Post 테이블

pk	title	content
1	첫글	안녕 나는 첫글
2	두번째글	안녕 난 두번째
3	세번째 글	일찍 끝내고 집가자!
NULL	NULL	NULL

Comment 테이블

pk	content	post
1	첫글인게 자랑이야?	1
2	아 내가 첫글 쓰더했는데 ..	1
3	세상은 일등근 기이하단다	2
4	첫 글 쓰 소감이 어때	1
5	무플방지위원회에서 나왔습니다	3
NULL	NULL	NULL





## 작성자 저장 (수정)

Auth > app > models.py

```
from django.contrib.auth.models import User

class Post(models.Model):
    title = models.CharField(max_length=200, default=None)
    content = models.TextField(default=None)
    author = models.ForeignKey(User, on_delete=models.CASCADE,
                               related_name='posts', null=True, default=None)

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



## 작성자 저장 (수정)

Auth > app > models.py

```
class Comment(models.Model):
    content = models.TextField(default=None)
    author = models.ForeignKey(User, on_delete=models.CASCADE,
                               related_name='comments', null=True, default=None)
```



# 작성자 저장 - 글작성

Auth > app > view.py

```
@login_required(login_url='/registration/login')
def new(request):
    if (request.method == 'POST'):
        new_post = Post.objects.create(
            title = request.POST['title'],
            content = request.POST['content'],
            author = request.user
        )
        return redirect('detail', new_post.pk)
    return render(request, 'new.html')
```



## 작성자 저장 - 댓글작성 (추가)

Auth > app > view.py

```
def detail(request):
    post = Post.objects.get(pk=post_pk)
    if (request.method == 'POST'):
        Comment.objects.create(
            post = post,
            content = request.POST['content'],
            author = request.user
        )
        return redirect('detail', post_pk)
    return render(request, 'detail.html', {'post': post})
```



## 작성자만 수정/삭제 가능 - 글 (추가)

Auth > app > templates > detail.html

```
<div>
    <div>{{ post.title }}</div>
    <div>{{ post.content }}</div>
</div>
<a href="{% url 'home' %}">홈으로</a>
{% if user.is_authenticated and post.author.pk == user.pk %}
    <a href="{% url 'edit' post.pk %}">수정하기</a>
    <a href="{% url 'delete' post.pk %}">삭제하기</a>
{% endif %}
```



## 작성자만 수정/삭제 가능 - 댓글

Auth > app > templates > detail.html

```
{% for comment in post.comments.all %}  
    <li>{{ comment.content }}</li>  
    {% if user.is_authenticated and comment.author.pk == user.pk %}  
        <a href="{% url 'delete_comment' post.pk comment.pk %}">댓글삭제</a>  
    {% endif %}  
{% endfor %}
```



# 로그인 시에만 댓글 작성 가능

Auth > app > templates > detail.html

```
{% if user.is_authenticated %}  
  <form method="POST">  
    {% csrf_token %}  
    <input type="text" name="content" placeholder="댓글을 입력해주세요" />  
    <button type="submit">댓글 쓰기</button>  
  </form>  
{% endif %}
```



# 오늘 할 일

- Issue 1** ~~회원가입 / 로그인 / 로그아웃 기능 구현~~
- Issue 2** ~~로그인 상태에 따라 다른 navbar~~
- Issue 3** ~~글의 작성자만 Post, Comment 수정, 삭제 권한 부여~~
- Issue 4** 카카오 API를 이용한 Social Login



Authentication

## 모듈 설치

NEXT[9].강단비

```
## 서버를 꺼줍니다 (control + c)
```

```
$ pipenv install django-allauth
```



# Setting 수정 (수정)

Auth > Auth > settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'app',
    'django.contrib.sites',
    'allauth',
    'allauth.account',
    'allauth.socialaccount',
    'allauth.socialaccount.providers.kakao'
]
```

(수정)

오타주의



# Setting 수정

Auth > Auth > settings.py

```
# social login 설정

AUTHENTICATION_BACKENDS = (
    'django.contrib.auth.backends.ModelBackend',
    'allauth.account.auth_backends.AuthenticationBackend',
)

SITE_ID = 1

LOGIN_REDIRECT_URL = '/'
```

오타주의



# 소셜로그인 path 지정 (추가)

Auth > Auth > urls.py

```
from Django.urls import path, include
```

(추가)

```
urlpatterns = [
    path('registration/signup', views.signup, name="signup"),
    path('registration/login', views.login, name="login"),
    path('registration/logout', views.logout, name="logout"),
    path('accounts/', include('allauth.urls')),
```

(추가)



## 확인

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

The screenshot shows the Django admin interface with the following sections:

- SITES**: A list titled "Sites" with two actions: "+ Add" and "Change".
- SOCIAL ACCOUNTS**: A list with three items: "Social accounts", "Social application tokens", and "Social applications", each with "+ Add" and "Change" actions.



Authentication

NEXT[9].강단비

# Sites 수정

Select site to change

ADD SITE +



Search

Action:

Go

0 of 1 selected



DOMAIN NAME



example.com

DISPLAY NAME

example.com

1 site



# Sites 수정

Change site

HISTORY

**Domain name:** http://127.0.0.1:8000

**Display name:** http://127.0.0.1:8000

**Delete** **Save and add another** **Save and continue editing** **SAVE**



# Social applications 추가

SITES		
Sites	<a href="#"> Add</a>	<a href="#"> Change</a>

SOCIAL ACCOUNTS		
Social accounts	<a href="#"> Add</a>	<a href="#"> Change</a>
Social application tokens	<a href="#"> Add</a>	<a href="#"> Change</a>
Social applications	<a href="#"> Add</a>	<a href="#"> Change</a>



# Kakao에 앱 생성

**Step 1** <https://developers.kakao.com/> 접속

**Step 2** 회원가입 / 로그인

**Step 3** 내 어플리케이션

**Step 4** 어플리케이션 추가하기



Authentication

# Kakao에 앱 생성

NEXT[9].강단비

내 애플리케이션 > 앱 설정 > 요약 정보

앱 설정

요약 정보

일반

비즈니스

앱 키

플랫폼

팀 관리

제품 설정

카카오 로그인

동의항목

간편가입

APP ID 463310 OWNER Web

앱 키

네이티브 앱 키

REST API 키

JavaScript 키

Admin 키

The screenshot shows the Kakao App Settings interface. On the left, there's a sidebar with various tabs like 'App Settings', 'General', 'Business', etc. The 'App Keys' tab is currently selected. On the right, under the 'App Keys' section, there are four categories: 'Native App Key', 'REST API Key', 'JavaScript Key', and 'Admin Key'. Each category has a yellow bar representing the key value, which is redacted in the image.



Authentication

# Kakao에 앱 생성

NEXT[9].강단비

내 애플리케이션 > 앱 설정 > 요약 정보

앱 설정

요약 정보

일반

비즈니스

앱 키

플랫폼

팀 관리

제품 설정

카카오 로그인

동의항목

간편가입

APP ID 463310 OWNER Web

앱 키

네이티브 앱 키

REST API 키

JavaScript 키

Admin 키

The screenshot shows the Kakao App Settings interface. On the left, there's a sidebar with various tabs like 'App Settings', 'General', 'Business', etc. The 'App Keys' tab is currently selected. On the right, under the 'App Keys' section, there are four categories: 'Native App Key', 'REST API Key', 'JavaScript Key', and 'Admin Key'. Each category has a yellow bar representing the key value, which is redacted in the image.



Authentication

# Kakao에 앱 생성

NEXT[9].강단비

내 애플리케이션 > 앱 설정 > 플랫폼

앱 설정  
요약 정보  
일반  
비즈니스  
앱 키  
**플랫폼**  
팀 관리

iOS

**iOS 플랫폼 등록**

Web

삭제      수정

제품 설정  
카카오 로그인  
동의항목  
간편가입

사이트 도메인      http://localhost:8000

• 카카오 로그인 사용 시 Redirect URI를 등록해야 합니다. [등록하러 가기](#)



# Kakao에 앱 생성 (추가)

일반  
비즈니스  
**앱 키**  
플랫폼  
팀 관리

제품 설정  
**카카오 로그인**  
동의항목  
간편가입  
카카오톡 채널  
개인정보 국외이전  
연결 끊기  
사용자 프로퍼티  
보안  
고급  
카카오링크  
카카오톡 채널

**카카오 로그인** ON

**활성화 설정**

상태 **ON**

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로 서비스를 시작할 수 있습니다.  
상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다.  
상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

**Redirect URI**

삭제 수정

Redirect URI  
http://localhost:8000/accounts/kakao/login/callback/  
http://127.0.0.0:8000/accounts/kakao/login/callback/

• 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)  
• REST API로 개발하는 경우 필수로 설정해야 합니다.



# Kakao에 앱 생성 (추가)

일반  
비즈니스  
앱 키  
**플랫폼**  
팀 관리

제품 설정  
카카오 로그인  
**동의항목**  
간편가입  
카카오톡 채널  
개인정보 국외이전  
연결 끊기  
사용자 프로퍼티  
보안  
고급  
카카오링크

카카오 로그인 **ON** 동의 화면 미리보기

**동의항목**

카카오 로그인으로 서비스를 시작할 때 동의 받는 항목을 설정합니다. 미리 보기를 통해 사용자에게 보여질 화면을 확인할 수 있습니다.  
사업자 정보를 등록하여 비즈 앱으로 전환하고 비즈니스 채널을 연결하면 권한이 없는 동의 항목에 대한 검수 신청을 할 수 있습니다.

**비즈니스 설정 바로가기**

**개인정보 보호**

항목 이름	ID	상태
프로필 정보(닉네임/프로필 사진)	profile	● 필수 동의 [수집] 설정
카카오톡 채널 추가 상태 및 내역	plusfriends	○ 권한 없음
카카오계정(이메일)	account_email	● 사용 안함 설정



# Kakao에 앱 생성

일반  
비즈니스  
앱 키  
플랫폼  
팀 관리

제품 설정  
**카카오 로그인**  
동의항목  
간편가입  
카카오톡 채널  
개인정보 국외이전  
연결 끊기  
사용자 프로퍼티  
**보안**  
고급  
카카오링크

카카오 로그인 **ON**

**Client Secret** 삭제

токن 발급 시, 보안을 강화하기 위해 Client Secret을 사용할 수 있습니다. (REST API인 경우에 해당)

코드	활성화 상태	사용함	설정
[REDACTED]			



Authentication

NEXT[9].강단비

# Admin에 kakao 등록

Provider: Kakao

Name: kakao

Client id: REST API KEY  
App ID, or consumer key

Secret key: CLIENT SECRET CODE  
API secret, client secret, or consumer secret

Key:

Sites:

Available sites	Chosen sites
<input type="text"/> Filter	<input checked="" type="text"/> http://127.0.0.1:8000
<input type="text"/>	<input type="text"/>



# Admin에 kakao 등록

Auth > app > templates > base.html

```
{% load static %}  
{% load socialaccount %}  
{% providers_media_js %}
```

```
<a class="nav-item" href="{% url 'login' %}>로그인</a>  
<a class="nav-item" href="{% provider_login_url 'kakao' %}>카카오 로그인</a>  
<a class="nav-item" href="{% url 'signup' %}>회원가입</a>
```



## 오늘 할 일

**Issue 1** ~~회원가입 / 로그인 / 로그아웃 기능 구현~~

**Issue 2** ~~로그인 상태에 따라 다른 navbar~~

**Issue 3** ~~글의 작성자만 Post, Comment 수정, 삭제 할 수 있어~~

**Issue 4** ~~카카오 API를 이용한 Social Login~~





# Assignment

**Issue 1** 자신이 만든 과제 블로그에 로그인 기능 구현

**Issue 2** 내가 쓴 글 모아 보는 페이지 구현

**다음 주 월요일까지 (2021-05-17)**

대충\_여러분의\_과제하는\_모습\_짤.jpg

