

4장. 로그인, 로그아웃 & 회원 가입



목 차

1

로그인 구현

2

로그아웃 구현

3

회원 가입

common 앱 생성하기

- common 앱 생성 후 초기 설정 작업

1. common 앱 생성하기

```
(mysite) C:\webproject\pyweb>python manage.py startapp common
```

2. config/setting.py 에 'common' 앱 등록하기

```
INSTALLED_APPS = [  
    'common.',  
    'board',  
    'pybo.apps.PyboConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
]
```

common 앱 생성하기

- common 앱 생성 후 초기 설정 작업

3. config/urls.py 경로 설정

```
from pybo import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('pybo/', include('pybo.urls')),
    path('common/', include('common.urls')),
]
```

4. common/urls.py 생성하기

```
app_name = 'common'

urlpatterns = [
]
```

로그인 구현하기

1. 네비게이션바 수정하기 – templates/navbar.html

```
<ul class="navbar-nav">
  <li class="nav-item">
    <a class="nav-link" href="{% url 'common:login' %}">로그인</a>
  </li>
</ul>
```

2. common/urls.py에 URL 매핑하기

```
from django.urls import path
from django.contrib.auth import views as auth_views

app_name = 'common'
urlpatterns = [
    path('login/', auth_views.LoginView.as_view(
        template_name='common/login.html'), name='login'),
]
```

제네릭뷰

로그인 구현하기

3. templates/common/login.html 생성

```
{% extends 'base.html' %}
{% block content %}
<div class="container">
  <div class="row my-3">
    <div class="col-4">
      <h4>로그인</h4>
    </div>
    <div class="col-8 text-right">
      <span>
        또는 <a href="{% url 'common:signup' %}">계정을 만드세요</a>
      </span>
    </div>
  </div>
  <form method="post">
    {% csrf_token %}
```

로그인 구현하기

3. templates/common/login.html 생성

```
{% include 'form_errors.html' %}
<div class="form-group">
    <label for="username">사용자ID</label>
    <input type="text" class="form-control" id="username" name="username"
        value="{% form.username.value|default_if_none:'' %}">
    <!-- name 속성값은 반드시 username으로 함-->
</div>
<div class="form-group">
    <label for="password">비밀번호</label>
    <input type="password" class="form-control" id="password" name="password"
        value="{% form.password.value|default_if_none:'' %}">
    <!-- name은 반드시 password로 함-->
</div>
<button type="submit" class="btn btn-primary">로그인</button>
</form>
</div>
{% endblock %}
```

로그인 구현하기

✓ 로그인 시 유효성 검사

- 입력값이 비어 있는 경우 오류 처리

Pybo

사용자 이름 필수 항목입니다.

비밀번호 필수 항목입니다.

사용자ID

비밀번호

로그인

로그인 구현하기

4. templates/form_errors.html 만들어 작성하기

```
{% if form.errors %}
    {% for field in form %}
        {% for error in field.errors %}
            <div class="alert alert-danger">
                <strong>{{ field.label }}</strong>
                {{ error }}
            </div>
        {% endfor %}
    {% endfor %}
{% endif %}
```

항목이 여러 개 이므로 중첩 for문을 사용한다.

로그인 구현하기

admin/12345로 로그인 해보기

Pybo 로그인

사용자ID

admin

비밀번호

.....

로그인

Page not found (404)

Request Method: GET

Request URL: http://127.0.0.1:8000/accounts/profile/

Using the URLconf defined in config.urls, Django tried these URL p

1. admin/
- 2.
3. pybo/
4. common/

5. 로그인 성공시 이동할 페이지 등록하기 – config/settings.py

```
# 로그인 로그아웃 성공 시 이동할 URL (루트 경로)
LOGIN_REDIRECT_URL = '/'
LOGOUT_REDIRECT_URL = '/'
```

auth 모듈

◆ django.contrib.auth 모듈

Django Documention > auth 로 검색

django.contrib.auth

This document provides API reference material for the components of Django's authentication system. For more details on the usage of these components or how to customize authentication and authorization see the [authentication topic guide](#).

User model

`class models.User`

Fields

`class models.User`

User objects have the following fields:

`username`

Attributes

`class models.User`

`is_authenticated`

Read-only attribute which is always **True** (as opposed to `request.user.is_authenticated` which is always **False**). This is a way to tell if the user has been authenticated and has permissions and doesn't check if the user is active or has been deleted. You can check this attribute on `request.user` to find out whether the user is authenticated.

auth 모듈

◆ django.contrib.auth 모듈

```
def login(request, user, backend=None):
```

```
    """
```

```
Persist a user id and a backend in the request. This way  
have to reauthenticate on every request. Note that data  
the anonymous session is retained when the user logs in.
```

```
    """
```

```
    session_auth_hash = ''
```

```
    if user is None:
```

```
        user = request.user
```

```
    if hasattr(user, 'get_session_hash'):
```

```
        session_auth_hash = user.get_session_hash()
```

```
def logout(request):
```

```
    """
```

```
Remove the authenticated user's ID from the request  
data.
```

```
    """
```

```
# Dispatch the signal before the user is logged out  
# chance to find out *who* logged out.
```

```
    user = getattr(request, 'user', None)
```

```
    if not getattr(user, 'is_authenticated', True):
```

```
        user = None
```

auth 모듈

- **User 클래스(객체)**

- User 객체는 인증시스템(Authentication System)의 핵심이다.
- 장고 인증은 암호 및 권한을 신중하게 구현하도록 발전했음

- **User 클래스의 주요 속성**

- Username
- Password
- Email
- First_name
- Last_name

auth 모듈

User model

```
class models.User
```

Fields

```
class models.User
```

User objects have the following fields:

username

Required. 150 characters or fewer. Usernames may contain alphanumeric, _, @, +, . and - characters.

The **max_length** should be sufficient for many use cases. If you need a longer length, please use a [custom user model](#).

If you use MySQL with the **utf8mb4** encoding (recommended for proper Unicode support), specify at most **max_length=191** because MySQL can only create unique indexes with 191 characters in that case by default.

```
create_user(username, email=None, password=None, **extra_fields) ¶
```

Creates, saves and returns a User.

The username and password are set as given. The domain portion of email is automatically generated and the returned User object will have is_active set to **True**.

If no password is provided, set_unusable_password() will be called.

auth 모듈

● User 모델 조회하기 – 장고 셸

```
>>> from django.contrib.auth.models import User
>>> User.objects.all()
<QuerySet [<User: admin>, <User: cloud>]>
>>> user1 = User.objects.get(id=1)
>>> user1.username
'admin'
>>> user1.password
'pbkdf2_sha256$260000$kMVupRiCQoA9S0iS8zIgYr$0QHYicYL/Wq69pHnne6uf5lTULr7G8/+pFSlKfrJl0Q='
>>> user2 = User.objects.get(username='cloud')
>>> user2.id
2
>>> user2.username
'cloud'
>>> user2.password
'pbkdf2_sha256$260000$PUaj2XYnC4g5QH99eam0n6$bjRvbIdNRyd+v8wLCPh1aZP7R2ENjktNGzARuguvezo='
```

auth 모듈

● 사용자 생성하기

```
>>> from django.contrib.auth.models import User
>>> user = User.objects.create_user('steve', 'haha@mbc.com', '1004')
>>> user.last_name = 'Jobs'
>>> user.save()
>>> user.username
'steve'
>>> user.password
'pbkdf2_sha256$260000$jv4Zquu7mECcSdBMihL4JC$VvVA6R4lufTYAIVqai+
```

id	password	last_login	is_superuser	username	last_name	email	is_staff	is_active	date_joined	first_name
...	필터	필터	필터	필터	필터	필터	필터	필터	필터	필터
1	pbkdf2_sha256\$260000\$VTQlIZ...	2021-07-15 14:25:26.659377	1	admin		test@test.com	1	1	2021-07-12 04:05:47.843006	
2	pbkdf2_sha256\$260000\$GYVhC...	NULL	0	haha		haha@mbc.com	0	1	2021-07-15 14:46:14.654352	
3	pbkdf2_sha256\$260000\$jv4Zq...	NULL	0	john	Lenon	haha@mbc.com	0	1	2021-07-15 14:46:51.545720	

auth 모듈

- 비밀번호 변경하기

```
>>> from django.contrib.auth.models import User
>>> u = User.objects.get(username='haha')
>>> u.set_password('haha3434')
>>> u.save()
```

Pybo 로그인

사용자ID	<input type="text" value="haha"/>
비밀번호	<input type="password" value="haha3434"/>
<input type="button" value="로그인"/>	

Pybo haha(로그아웃)	
번호	제목
1	haha
2	장마
3	웹 기초기술
4	장고로 만들어진 유명

auth 모듈

- 관리자 페이지에서 보기

Django 관리

홈 > 인증 및 권한 > 사용자(들)

PYBO	
Answers	+ 추가
Questions	+ 추가
인증 및 권한	
그룹	+ 추가
사용자(들)	+ 추가

변경할 사용자 선택

Q 검색

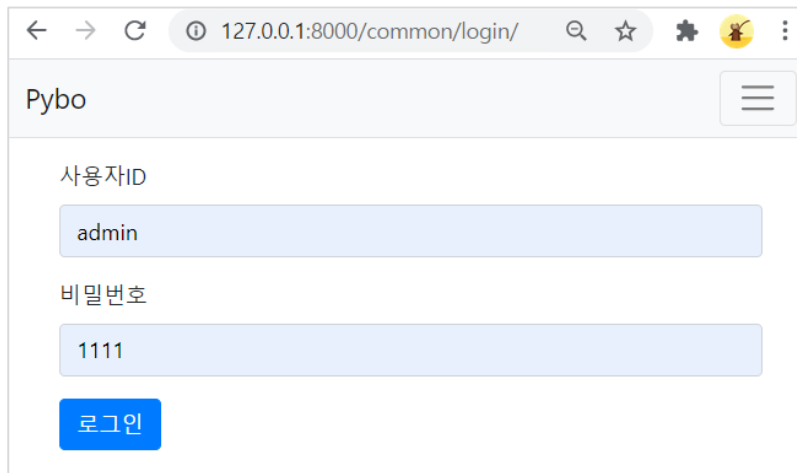
액션: 실행 3 중 아무것도

<input type="checkbox"/>	사용자 이름	이메일 주소
<input type="checkbox"/>	admin	test@test.com
<input type="checkbox"/>	haha	haha@mbc.com
<input type="checkbox"/>	john	haha@mbc.com

3 사용자(들)

로그아웃 구현하기

● 로그아웃 구현하기



A screenshot of a web browser window. The address bar shows '127.0.0.1:8000/common/login/'. The page title is 'Pybo'. There are two input fields: '사용자ID' (User ID) with the value 'admin' and '비밀번호' (Password) with the value '1111'. Below the fields is a blue button labeled '로그인' (Login).

아이디 : admin
비밀번호 : 1111



A screenshot of a web page titled 'Pybo (로그아웃)'. It features a table with two columns: '번호' (Number) and '제목' (Title). The table contains four rows of data. Below the table is a blue button labeled '질문 등록하기' (Register Question).

번호	제목
1	장마
2	웹 기초기술
3	장고로 만들어진 유명 사이트가 있나요?
4	Django Model Question

로그아웃 구현하기

● 로그아웃 구현하기

1. 네비게이션바 수정하기 – templates/navbar.html

```
<!-- 로그인 메뉴 -->
<div class="collapse navbar-collapse flex-grow-0" id="navbarNav">
  <ul class="navbar-nav">
    <li class="nav-item">
      {% if user.is_authenticated %} <!-- 로그인(인증) 되었다면 -->
        <a class="nav-link" href="{% url 'common:logout' %}">
          {{ user.username }} (로그아웃)
        </a>
      {% else %}
        <a class="nav-link" href="{% url 'common:login' %}">로그인</a>
      {% endif %}
    </li>
  </ul>
</div>
```

로그아웃 구현하기

- 로그아웃 구현하기

- 2. 로그아웃 URL 매핑하기 – common/urls.py

```
urlpatterns = [  
    path('login/', auth_views.LoginView.as_view(  
        template_name='common/login.html'), name='login'),  
    path('logout/', auth_views.LogoutView.as_view(), name='logout'),  
]
```

회원 가입 구현하기

- 회원 가입

1. 회원가입 링크 추가하기 – common/signup.html

```
<div class="container my-3">
  <div class="row">
    <div class="col-4">
      <h4>로그인</h4>
    </div>
    <div class="col-8 text-right">
      <span>또는 <a href="{% url 'common:signup' %}">계정을 만드세요</a></span>
    </div>
  </div>
</div>
```

회원 가입 구현하기

2. 회원가입 URL 매핑 추가하기 – common/urls.py

```
from . import views
urlpatterns = [
    ....
    path('signup/', views.signup, name='signup'),
]
```

회원 가입 구현하기

3. **forms.py** 만든후 UserForm 생성하기 – common/forms.py

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

class UserForm(UserCreationForm):
    email = forms.EmailField(label="이메일")

    class Meta:
        model = User
        fields = ('username', 'email')
```


회원 가입 구현하기

4. 회원가입을 위한 signup 함수 정의 – common/views.py

```
def signup(request):  
    #회원 가입  
    if request.method == "POST":  
        form = UserForm(request.POST)  
        if form.is_valid():  
            form.save()  
            # 가입후 자동 로그인  
            username = form.cleaned_data.get('username') #사용자ID  
            password1 = form.cleaned_data.get('password1') #비밀번호  
            user = authenticate(username=username, password=password1)  
            login(request, user) #로그인  
            return redirect('board:index')  
        else:  
            form = UserForm()  
            context = {'form': form}  
            return render(request, 'common/signup.html', context)
```

회원 가입 구현하기

UserCreationForm

태그	설명
username	사용자 이름
password1	비밀번호1
password2	비밀번호2(비밀번호 확인)

```
class UserCreationForm(forms.ModelForm):  
    """  
    A form that creates a user, with no privileges, from the given username  
    password.  
    """  
    error_messages = {  
        'password_mismatch': _('The two password fields didn't match.'),  
    }  
    password1 = forms.CharField(  
        label=_("Password"),  
        strip=False,  
        widget=forms.PasswordInput(attrs={'autocomplete': 'new-password'}),  
        help_text=password_validation.password_validators_help_text_html(),  
    )
```

Reader Mode

회원 가입 구현하기

5. 회원가입 템플릿 만들기 – common/signup.html

Pybo

로그인

회원 가입

또는 [로그인 하세요](#)

사용자 이름

비밀번호

비밀번호 확인

이메일

생성하기

회원 가입 구현하기

5. 회원가입 템플릿 만들기 – common/signup.html

```
{% extends 'base.html' %}
{% block content %}
<div class="container my-3">
  <div class="row my-3">
    <div class="col-4">
      <h4>회원가입</h4>
    </div>
    <div class="col-8 text-right">
      <span>또는 <a href="{% url 'common:login' %}">로그인 하세요</a> </span>
    </div>
  </div>
  <form method="post" class="post-form">
    {% csrf_token %}
    {% include 'form_errors.html' %}
    <div class="form-group">
      <label for="username">사용자 이름</label>
      <input type="text" class="form-control" name="username" id="username"
        value="{{ form.username.value|default_if_none:'' }}">
    </div>
  </form>
</div>
```

회원 가입 구현하기

5. 회원가입 템플릿 만들기 – common/signup.html

```
<div class="form-group">
  <label for="password1">비밀번호</label>
  <input type="password" class="form-control" name="password1" id="password1"
    value="{{ form.password1.value|default_if_none:'' }}">
</div>
<div class="form-group">
  <label for="password2">비밀번호 확인</label>
  <input type="password" class="form-control" name="password2" id="password2"
    value="{{ form.password2.value|default_if_none:'' }}">
</div>
<div class="form-group">
  <label for="email">이메일</label>
  <input type="text" class="form-control" name="email" id="email"
    value="{{ form.email.value|default_if_none:'' }}">
</div>
<button type="submit" class="btn btn-primary">생성하기</button>
</form>
```

회원 가입 구현하기

회원가입

비밀번호 확인 비밀번호가 너무 일상적인 단어입니다.

사용자 이름

giyong

비밀번호

.....

비밀번호 확인

.....

이메일

kiyongee2@gmail.com

생성하기

회원가입

사용자 이름 해당 사용자 이름은 이미 존재합니다.

비밀번호 확인 비밀번호가 일치하지 않습니다.

사용자 이름

giyong

비밀번호

.....

비밀번호 확인

.....

이메일

test@test.kr

생성하기

글쓴이 표시하기

- 글쓴이(author) 추가하기

1. Question 모델에 author 필드 추가하기 – board/models.py

```
class Question(models.Model):  
    author = models.ForeignKey(User, on_delete=models.CASCADE)  
    subject = models.CharField(max_length=200)  
    content = models.TextField()  
    create_date = models.DateTimeField()
```

글쓴이 표시하기

2. Question 모델에 author 필드 추가하기 – board/models.py

```
(mysite) c:\w\projects\wpybo2>python manage.py makemigrations
You are trying to add a non-nullable field 'author' to question without a default;
Please select a fix:
  1) Provide a one-off default now (will be set on all existing rows with a null value)
  2) Quit, and let me add a default in models.py
Select an option: 1
Please enter the default value now, as valid Python
The datetime and django.utils.timezone modules are available, so you can do e.g.
Type 'exit' to exit this prompt
>>> 1
Migrations for 'pybo':
  pybo\migrations\0002_question_author.py
  - Add field author to question
```

null값 대신에 강제로 임의계
정을 추가하는 방법

최초 생성한 슈퍼 유저의 id값
을 뜻한다.

3. migrate 명령 실행하기

글쓴이 표시하기

- Answer 모델 수정하기

1. Answer 모델에 author 필드 추가하기 – board/models.py

```
class Answer(models.Model):  
    author = models.ForeignKey(User, on_delete=models.CASCADE)  
    question = models.ForeignKey(Question, on_delete=models.CASCADE)  
    # 외래키 제약조건 무시하고 연쇄 삭제 됨  
    content = models.TextField()  
    create_date = models.DateTimeField()
```

글쓴이 표시하기

2. Answer 모델에 author 필드 추가하기 – board/models.py

```
(mysite) c:\w\projects\wpybo2>python manage.py makemigrations
You are trying to add a non-nullable field 'author' to answer without a default;
Please select a fix:
  1) Provide a one-off default now (will be set on all existing rows with a null value)
  2) Quit, and let me add a default in models.py
Select an option: 1
Please enter the default value now, as valid Python
The datetime and django.utils.timezone modules are available, so you can do e
Type 'exit' to exit this prompt
>>> 1
Migrations for 'pybo':
  pybo\migrations\0003_answer_author.py
    - Add field author to answer
```

3. migrate 명령 실행하기

글쓴이 표시하기

- author 필드 적용하기

1. 질문 등록 함수 수정하기 – board/views.py

```
def question_create(request):  
    ...  
    if form.is_valid():  
        question = form.save(commit=False)  
        question.author = request.user
```

2. 답변 등록 함수 수정하기 – board/views.py

```
def answer_create(request, question_id):  
    ...  
    if form.is_valid():  
        answer = form.save(commit=False)  
        answer.author = request.user
```

글쓰기 표시하기

- 로그아웃 상태에서 글쓰기 오류 및 처리

로그인한 유저만 글쓰기 가능 – board/views.py

```
ValueError at /pybo/question/create/  
Cannot assign "<SimpleLazyObject: <django.contrib.auth.models.AnonymousUser  
instance.  
  
Request Method: POST  
Request URL: http://127.0.0.1:8000/pybo/question/create/  
Django Version: 3.2  
Exception Type: ValueError
```

```
from django.contrib.auth.decorators import login_required  
@login_required(login_url='common:login')  
def question_create(request):  
    ...  
  
@login_required(login_url='common:login')  
def answer_create(request, question_id):  
    ...
```

글쓴이 표시하기

- 로그아웃 상태에서 답변등록 비활성화

답변 등록 비활성화하기 – board/detail.html

```
<div class="form-group">  
  <textarea name="content" id="content" rows="10" class="form-control"  
    {% if not user.is_authenticated %}  
      disabled  
    {% endif %}></textarea>  
</div>
```

1개의 답변이 있습니다.

비타민 C가 풍부하데요.

2021년 7월 18일 5:50 오전

답변 등록

글쓴이 표시하기

번호	제목	글쓴이	작성일시
1	브로콜리2	coli	2021년 7월 18일 5:55 오전
2	브로콜리	coli	2021년 7월 18일 5:50 오전
3	코로나	admin	2021년 7월 16일 5:44 오전

브로콜리2

브로콜리의 원산지는 어디인가여?

coli
2021년 7월 18일 5:55 오전

2개의 답변이 있습니다.

지중해 연안입니다.

coli
2021년 7월 18일 5:55 오전

브로콜리는 비타민 C가 풍부해요

giyong
2021년 7월 18일 5:55 오전

글쓴이 표시하기

- 질문 목록 화면에 글쓴이 표시하기

```
<thead class="thead-dark">
  <tr>
    <th>번호</th><th>제목</th><th>글쓴이</th><th>작성일시</th>
  </tr>
</thead>
<tbody>
  {% if question_list %}
    {% for question in question_list %}
      <tr>
        <td>{{ forloop.counter }}</td>
        <td><a href="{% url 'board:detail' question.id %}">
          {{ question.subject }}</a>
        </td>
        <td>{{ question.author }}</td>
        <td>{{ question.create_date }}</td>
      </tr>
    {% endfor %}
  {% else %}
    <tr>
      <td colspan="4">질문 목록이 없습니다.</td>
    </tr>
  {% endif %}
</tbody>
```

글쓴이 표시하기

- 질문/답변 상세 화면에 글쓴이 표시하기

```
<div class="d-flex justify-content-end">  
  <div class="badge badge-light p-2 text-left">  
    <div class="mb-2">{{ question.author }}</div>  
    <div>{{ question.create_date }}</div>  
  </div>  
</div>
```

```
<div class="d-flex justify-content-end">  
  <div class="badge badge-light p-2 text-left">  
    <div class="mb-2">{{ answer.author }}</div>  
    <div>{{ answer.create_date }}</div>  
  </div>  
</div>
```