

Django

- 깃 소스 코드 참고(Django_Works)

<Django의 개요>

Spring Boot와 같이 ORM 기반의 프로그램

Django는 Python으로 작성된 오픈 소스 웹 프레임워크

MVC 패턴 지원, ORM 지원, 인증 및 권한 관리, 템플릿 시스템, 미디어 파일 관리 등의 특징이 있습니다.

<Django 설치 방법>

pip install django - Django 설치(오류가 나면 관리자모드로 설치해주세요.)

pip install mysqlclient - Django에서 DB(MySQL)연결을 하기 위한 설치

<프로젝트 생성>

django-admin startproject [project명]

예시> django-admin startproject myDjango01

myDjango01로 이동

<프로젝트 앱 실행>

python manage.py startapp myapp01

dir/w - 디렉토리 확인

<settings.py 설정>

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'myapp01'  
]
```

settings.py에서 INSTALLED_APPS에 myapp01 읽어올 수 있도록 추가

<Schema 추가>

django.db - MySQL에 스키마 추가

<설정>

```
DATABASES = {  
    'default': {
```

```
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

에서

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'djangodb',
        'USER' : 'root',
        'PASSWORD' : 'root',
        'HOST' : 'localhost',
        'PORT' : '3306',
    }
}
```

로 변경

```
LANGUAGE_CODE = 'ko'
TIME_ZONE = 'Asia/Seoul'
USE_I18N = True
USE_L10N = True
USE_TZ = False
```

로 설정

```
STATIC_URL = 'static/' - static은 이미지 등을 저장하는 폴더(정적파일 관리폴더)
STATIC_URL = '/myapp01/static/' 으로 변경
python manage.py makemigrations - 변경점 확인
python manage.py migrate - 마이그레이션
python manage.py runserver - 서버 구동
```

템플릿 만들기

```
templates 폴더 생성(기본적으로는 templates의 폴더를 찾아간다)
borad 폴더 생성 -> write.html 생성 -> !입력(HTML 뼈대 자동완성)
-> Body에 <h3>Insert</h3>추가
```

urls.py에

import에 추가

```
from myapp01 import views
```

urlpatterns에

```
path('write_form/', views.write_form)
```

추가

views.py에

```
def write_form(request):
```

```
    return render(request, 'board/write.html')
```

추가해줘야 함

```
from django.views.decorators.csrf import csrf_exempt 추가
```

@csrf_exempt - csrf 사용 안함 의미

UPLOAD_DIR = 'C:/Django_Works/upload' 업로드 경로 잡아주기

@csrf_exempt 아래에

```
def insert(request):
```

```
    fname = ""
```

```
    fsize = 0
```

```
    if 'file' in request.FILES:
```

```
        file = request.FILES['file']
```

```
        fsize = file.size
```

```
        fname = file.name
```

```
        fp = open('%s%s' %(UPLOAD_DIR, fname), 'wb')
```

```
        for chunk in file.chunks():
```

```
            fp.write(chunk)
```

```
        fp.close()
```

```
    return render(request, 'board/write.html')를 추가
```

<models.py에 추가> - Entity 추가

python manage.py makemigrations - 변경점 확인

python manage.py migrate - 마이그레이션

완성 후 써주기

<models.py> - Entity 추가 작업

```
from django.db import models
```

```
from datetime import datetime
```

```
# Create your models here.
```

```
class Board(models.Model):
```

```
    idx = models.AutoField(primary_key=True)
```

```
writer = models.CharField(null=False, max_length=50)
title = models.CharField(null=False, max_length=200)
content = models.TextField(null=False)
hit = models.IntegerField(default=0)
post_date = models.DateTimeField(default=datetime.now, blank=True)
filename = models.CharField(null=True, blank=True, default="", max_length=500)
filesize = models.IntegerField(default=0)
down = models.IntegerField(default=0)
```

```
# 조회수 증가
```

```
def hit_up(self):
    self.hit += 1
```

```
# 파일 다운로드 수 증가
```

```
def down_up(self):
    self.down += 1
```

```
<views.py>
```

```
from django.shortcuts import render, redirect
from django.views.decorators.csrf import csrf_exempt
from myapp01.models import Board
# Create your views here.
```

```
UPLOAD_DIR = 'C:/Django_Works/upload/'
```

```
# write_form
```

```
def write_form(request):
    return render(request, 'board/write.html')
```

```
# insert
```

```
@csrf_exempt
```

```
def insert(request):
    fname = ""
    fsize = 0
    if 'file' in request.FILES:
        file = request.FILES['file']
        fsize = file.size
        fname = file.name
        fp = open('%s%s' %(UPLOAD_DIR, fname), 'wb')
        for chunk in file.chunks():
            fp.write(chunk)
```

```
fp.close()
```

```
# DTO 정의
```

```
    dto = Board(writer=request.POST['writer'],
                 title=request.POST['title'],
                 content=request.POST['content'],
                 filename=fname,
                 filesize=fsize
                )
    dto.save()
```

```
    return redirect("/list/")
```

```
# list 전체보기(기본)
```

```
def list(request):
    render(request, 'board/list.html')
```

파일 삽입까지 확인해주기

templates에 list.html 추가

<body>에

```
<h3>LIST</h3>
```

```
<table border="1">
```

```
<thead>
```

```
<tr>
```

```
<th>번호</th>
```

```
<th>제목</th>
```

```
<th>작성자</th>
```

```
<th>작성일</th>
```

```
<th>조회수</th>
```

```
<th>첨부파일</th>
```

```
<th>파일이미지</th>
```

```
<th>다운로드</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
{% for board in boardList %}
```

```
<tr>
```

```
<td>{{board.idx}}</td>
```

```
<td>{{board.title}}</td>
```

```

        <td>{{board.writer}}</td>
        <td>{{board.post_date}}</td>
        <td>{{board.hit}}</td>
        <td>{{board.filename}}</td>
        <td>파일이미지</td>
        <td>{{board.down}}</td>
    </tr>
{% endfor %}
</tbody>
</table>

```

추가

{% for board in boardList %}, {% endfor %} - Django에서의 반복문 구문

{% load static %} - Bootstrap, JQuery 등 로컬 파일 불러들이는 문법, 첫줄에 적어준다.

{% if board.filesize > 0 %}~{% endif %} - static에 있는 파일을 불러오는 문법

from django.http.response import JsonResponse(JSON 객체 불러오는 것)

def list(request):

page = request.GET.get('page',1)

word = request.GET.get('word', '')

field = request.GET.get('field', 'title')

#count

if field=='all':

```

    boardCount = Board.objects.filter(Q(writer__contains=word)|
                                       Q(title__contains=word)|
                                       Q(content__contains=word)).count()

```

elif field == 'writer':

```

    boardCount = Board.objects.filter(Q(writer__contains=word)).count()

```

elif field =='title':

```

    boardCount = Board.objects.filter(Q(title__contains=word)).count()

```

elif field =='content':

```

    boardCount = Board.objects.filter(Q(content__contains=word)).count()

```

else:

```

    boardCount = Board.objects.all().count

```

<Q 연산자>

Q객체 (Q연산자)란 장고에서 제공하는 기능으로 SQL로는 WHERE AND, OR, NOT 조건을 추가해주는 것이다.

Q객체를 활용하면 복잡한 필터링을 편하게 실행할 수 있다.

<삭제, 댓글 추가, 게시판 추가>

import math (반올림, 버림 사용)

STATICFILES_DIRS, BASE_DIR 이용하여 경로 잡기

<header, footer를 사용한 HTML5와 같은 형태 만들기>

<!-- 기본 템플릿 안에 삽입 될 내용 start-->

{% block content %}

{% endblock %}

<!-- JS start -->

{% block script %}

{% endblock %}

[Django 활용(ORM 기반)]

- 회원가입 폼 활용하기(django.contrib.auth 활용)

- 유효성검사

- 페이지징

<슈퍼유저 만들기>

python manage.py createsuperuser

사용자 이름 (leave blank to use 'admin'):

이메일 주소: admin@admin.com

Password: admin

Password (again): admin

비밀번호가 사용자 이름과 너무 유사합니다.

비밀번호가 너무 짧습니다. 최소 8 문자를 포함해야 합니다.

비밀번호가 너무 일상적인 단어입니다.

Bypass password validation and create user anyway? [y/N]: y

Superuser created successfully.

Django 관리

사이트 관리

인증 및 권한

그룹

+ 추가 ✎ 변경

사용자(들)

+ 추가 ✎ 변경

최근 활동

나의 활동

이용할 수 없습니다.

슈퍼유저 생성 후 localhost:8000/admin 입력하고 나서 이와 같은 창이 나타난다.

pip install konlpy (한국어 자연어 처리를 위한 라이브러리(한글 형태소 분석 라이브러리) JRE 필요)

{% get_static_prefix %} static을 참조하여 파일을 가져올 때 사용

pip install pytagcloud (단어 클라우드 만드는 라이브러리)

pip install pygame (프로토타이핑 라이브러리)

pip install simplejson(JSON 데이터 다루는 라이브러리)

폴더(C:\Python\Python311\Lib\site-packages\pytagcloud\fonts)에 폰트 다운로드해서 폰트 삽입하기

melon chart 1~10위까지 뽑아보기

pip install folium(지도 시각화 하는 라이브러리)

folium 활용 지도 띄우기

settings.py에서 TEMPLATE_DIR = os.path.join(BASE_DIR, 'myapp03/templates') 경로지정 추가

import os(운영 체제 인터페이스 표준 모듈 라이브러리)

from myDjango03.settings import TEMPLATE_DIR 추가

Pandas(groupby) (데이터를 특정 기준으로 그룹화하여 분석하는 기능이 groupby의 특징이다.)

font_manager(Python의 기본 폰트 관리 모듈)

rc(Python의 설정 파일, Python 프로그램의 기본 설정을 지정)

annotate는 QuerySet에 추가적인 필드를 추가하는 기능이다.

django.db.models.aggregates 쿼리셋 집계 모듈이다.(다양한 집계 함수를 제공)