

오픈소스 소프트웨어 프로젝트 회의록

일시	2024.10.03																								
회의장소	투썸플레이스 충무로역점																								
참석자명단	고윤건, 정가경, 허재원, 현광수																								
회의 주제	OSSP 개발을 위한 환경 설정 및 데이터베이스 연동																								
회의 내용	<p>1. 개발 환경</p> <p><Python env></p> <ul style="list-style-type: none"> - 개발 환경: Pycharm professional 2024.2.3. - 설치 패키지 (이후 추가될 예정) <table> <tr><td>Python</td><td>3.11.9</td></tr> <tr><td>asgiref</td><td>3.8.1</td></tr> <tr><td>Django</td><td>4.1.13</td></tr> <tr><td>django</td><td>1.3.6</td></tr> <tr><td>dnspython</td><td>2.6.1</td></tr> <tr><td>pip</td><td>23.2.1</td></tr> <tr><td>pymongo</td><td>3.12.1</td></tr> <tr><td>pytz</td><td>2024.2</td></tr> <tr><td>setuptools</td><td>68.2.0</td></tr> <tr><td>sqlparse</td><td>0.2.4</td></tr> <tr><td>tzdata</td><td>2024.2</td></tr> <tr><td>wheel</td><td>0.41.2</td></tr> </table> <p><주요 패키지 간단 설명></p> <ul style="list-style-type: none"> - Django <ul style="list-style-type: none"> • 파이썬으로 작성된 고수준 웹 프레임워크 • 웹 애플리케이션 개발에 필요한 많은 기능들을 내장하고 있어, 빠르고 효율적으로 개발할 수 있음 • 모델, 템플릿, 뷰, URL 라우팅 등 주요 컴포넌트를 제공 • 강력한 ORM(Object Relational Mapping) 시스템을 통해 데이터베이스와 상호작용할 수 있음 - django <ul style="list-style-type: none"> • Django의 ORM과 MongoDB를 연결해주는 패키지 • Django에서 MongoDB를 사용하기 위해서는 기본적으로 SQL을 기반으로 한 	Python	3.11.9	asgiref	3.8.1	Django	4.1.13	django	1.3.6	dnspython	2.6.1	pip	23.2.1	pymongo	3.12.1	pytz	2024.2	setuptools	68.2.0	sqlparse	0.2.4	tzdata	2024.2	wheel	0.41.2
Python	3.11.9																								
asgiref	3.8.1																								
Django	4.1.13																								
django	1.3.6																								
dnspython	2.6.1																								
pip	23.2.1																								
pymongo	3.12.1																								
pytz	2024.2																								
setuptools	68.2.0																								
sqlparse	0.2.4																								
tzdata	2024.2																								
wheel	0.41.2																								

	<p>ORM을 사용하는데, Django는 이를 MongoDB와 호환되도록 변환해주는 역할</p> <ul style="list-style-type: none"> Django의 ORM을 그대로 사용하면서도 MongoDB와 상호작용할 수 있음 <p>- pymongo</p> <ul style="list-style-type: none"> 파이썬에서 MongoDB와 직접적으로 통신하기 위한 공식 클라이언트 라이브러리 Django의 ORM을 사용하지 않고, 더 낮은 수준에서 MongoDB에 쿼리하고 데이터를 처리하고자 할 때 PyMongo를 사용 이는 MongoDB의 기본 기능을 최대한 활용하려는 경우에 유용 <h2>2. Django 프로젝트 생성 및 앱 생성</h2> <h3><Django 설치 및 프로젝트 생성></h3> <p>파이썬 개발도구 Terminal에서 실행</p> <ol style="list-style-type: none"> ① pip install django ② django-admin startproject config . ③ python manage.py runserver ④ 웹브라우저에서 http://127.0.0.1:8000 으로 접속해서 테스트 ⑤ 웹브라우저가 실행이 잘 된다면 django-admin startproject config . 입력해서 만든 폴더 확인 <p>config 폴더: 프로젝트의 설정 파일과 웹 서버 실행을 위한 파일들이 들어 있음</p> <ul style="list-style-type: none"> - __init__.py: 파이썬 2.X 버전과 호환을 위해 만들어진 파일, 여러 폴더에 생성됨, 지워도 무관 - asgi.py wsgi.py: 웹서버에 배포할 때 설정파일들을 연결해 주는 파이썬 파일 - settings.py: 프로젝트에 다양한 설정에 관한 내용이 들어있는 파일 - urls.py: 하나의 프로젝트에는 여러개의 urls 파일이 만들어지고 config 안의 urls 파일은 최초로 탐색되는 기준 urls 파일. 기준 urls 파일은 settings 파일에서 변경 가능. url에 접속했을 때 어떻게 처리할 것인지를 입력하는 파일. <p>config 폴더의 핵심 파일: settings.py, urls.py</p> <p>venv 폴더</p> <ul style="list-style-type: none"> - manage.py: 장고의 다양한 명령어를 실행하기 위한 파일. 수정할 필요 없음 - db.sqlite3: 로컬 DB 파일, 비어있음 <h3><Django App></h3> <p>App: 프로젝트를 이루는 작은 단위</p> <ul style="list-style-type: none"> - Django는 하나의 프로젝트 안에 여러 개의 앱이 포함됨 - 프로젝트를 하나의 서비스라고 하면, 앱은 그 서비스 기능을 큰 틀로 나눈 것이라고 할 수 있음 - 꼭 여러 앱으로 서비스 구현할 필요는 없지만, 하나의 앱만으로 서비스를 구현하면 초기 구현뿐만 아니라 유지보수가 어려워짐
--	---

- 그래서 큰 기능은 여러 앱으로 나눠서 구현하는 것이 좋음

① python manage.py startapp [앱이름]

② 앱을 생성하면 앱이름으로 폴더가 생기고 그 안에 여러 파일 생성

app의 파일

- admin.py: 관리자가 접속하면 보이는 화면, 내장돼 있음

- apps.py: 앱을 프로젝트에 등록하는 기능을 함

- models.py : 장고 DB 관련된 파일 / DB 사용계획, 정의, 연결 등의 다양한 설정들을 함.

- tests.py : 테스트를 위한 파이썬 파일

- views.py : 실질적으로 파이썬 코드가 실행이 되는 부분, 클래스형, 함수형 모두 가능.

- urls.py : 폴더 안에 없지만, 추가로 생성해서 사용해야 하는 파일

- migrations/ : 파이썬 모듈로 작동하는 폴더, 데이터 베이스 스키마 관련

app 폴더의 핵심 파일: models.py, views.py

<App 만들어보기(예시)>

① [앱이름]/views파일에서 함수 만들기

- func1이라는 함수 생성

- 이 함수는 “func1 함수 실행”이라고 출력하고 웹에서 “Hello Django World”라고 응답

- HttpResponse: html 파일을 사용하지 않고 클라이언트에게 response를 반환하는 가장 기본적인 메소드 (함수)

② config/urls.py 파일에 url 추가

- path('test/', [앱이름].views.func1),

- test/: 사용자가 접속할 url 의미

- [앱이름].views.func1): [앱이름] 폴더의 views.py 파일에 func1 함수를 실행한다는 의미

③ python manage.py runserver

④ 웹 브라우저에서 http://127.0.0.1:8000/test/으로 접속해서 확인

3. mongoDB 연동하기

<mongoDB에 database 생성>

① mongoDB 등록 및 접속

② database create 후 Browse Collections 들어가기

③ creat database 누른 후 생성할 DB 이름 생성

<django에서 mongoDB 연동>

- ① pip install djongo
- ② settings.py에서 DATABASES를 변경

```

DATABASES = {
    'default': {
        'ENGINE': 'djongo',
        'NAME': 'your-db-name',
        'ENFORCE_SCHEMA': False,
        'CLIENT': {
            'host': 'mongodb+srv://<username>:<password>@<atlas
cluster>/<myFirstDatabase>?retryWrites=true&w=majority'
        }
    }
}

```

- your-db-name: 연동하기 위해 만든 mongoDB의 database명
 - host: MongoDB에서 Connect에 들어간 후 Driver에서 3. Add your connection string into application code에 적힌 값을 입력
 - password: db access id와 pw 생성할 때 만들었던 password
- ③ 에러가 뜨는 경우 pip install pytz
 - ④ 그래도 에러가 발생하는 경우 pymongo 버전을 환경에 맞게 세팅
 - ⑤ 문제가 없는 경우 python manage.py makemigrations
 - No changes detected가 뜬 경우 성공
 - ⑥ python manage.py migrate
 - ⑦ mongoDB에 들어가서 확인하면 데이터베이스가 들어가 있는 것을 확인