

Date: 0310

Django database

Model

- 웹 데이터를 구조화하고 조작하기 위한 도구
 - 단일한 데이터에 대한 정보를 가짐
 - 데이터의 필수적 필드들과 동작들을 포함
 - 모델은 데이터베이스를 다루는 도구 (모델 != 데이터베이스)
 - 모델과 데이터베이스 테이블 1:1로 매핑

DataBase

- 체계화된 데이터의 모임
- query
 - 데이터를 조회하기 위한 명령어
 - 조건에 맞는 데이터를 추출 또는 조작한다

DataBase의 기본구조

- schema
 - 데이터베이스에서의 자료의 구조, 표현방법, 관계를 나타낸 구조
- 테이블(table)
 - 열 == 필드/컬럼/속성
 - 행 == 레코드/행/튜플
 - pk(기본키): 각 행의 고유값으로 단 하나의 값으로 관계설정과 데이터베이스 관리시 매우 유용하게 사용된다.
 -

ORM(Object-Relational-Mapping)

객체 지향 언어를 이용하여 호환되지 않는 두 시스템(django-sql)사이 데이터를 변환하는 프로그래밍 기술이다.

- 장점
 - SQL을 모르지만 DB를 조작가능
 - 객체 지향적 접근을 통해서 높은 생산성(현대 웹프레임워크의 요점!!)
- 단점
 - ORM만으로는 완전한 서비스를 구현하기 어려움
- 서술형: 우리는 db를 객체로 조작하기 위해 orm을 사용한다!

Migrations

- 장고가 model에 생긴 변화를 반영하는 방법
- makemigrations
 - 변경 모델을 기반으로 마이그레이션(설계도) 작성
- migrate
 - 마이그레이션을 실제 db에 반영
- sqlmigrate
 - 마이그레이션에 대한 sql 구문을 보기 위해 사용
- showmigrations
 - 프로젝트 전체의 마이그레이션 상태를 확인(migrate 적용 여부 확인)

Database API

- db를 편하게 조작하기 위한 도구
- database-abstract API, database-access API 로도 부름
- 사용법
 - Class_name.Manager.Querset_API
 - ex) article.objects.all()
- manager
 - django 모델에 데이터베이스 query 작업이 제공되는 인터페이스
- queryset
 - 데이터베이스로부터 전달받은 객체의 목록
 - 조회, 필터, 정렬 등 수행가능

crud

- create
 - 객체를 데이터베이스의 저장
 - save()를 안하면 객체의 id값을 알 수 없음
- read
 - method that return new querysets
 - method that do not return querysets
 - field lookups : 조건 시 특정 조건을 적용시키기 위해서 키워드 인수를 사용 (get,filter,exclude)
 - get -> 해당 객체가 없으면 DoesNotExist 에러발생
 - get-> 해당 객체가여러개면 MultipleObjectsReturned 에러 발생
 - not null 특징이나 unique 속성을 가질 때만 사용(== pk일때만 사용)
 - filter -> pk__gt =1 pk가 1보다 큰 값들을 가져온다
- update
- delete

Admin

- 서버관리자 활용하기 위한 페이지
- record 생성 여부 확인이 쉽고, 직접 추가도 가능
- django.contrib.auth 모듈에서 제공
- admin.py에 모델을 등록해서 사용