


Create, Read, Update, Delete 기능의 흐름

url은 식별

1. 컨트롤러가 요청 받음
2. 서비스에 요청(주소 + id)을 보냄

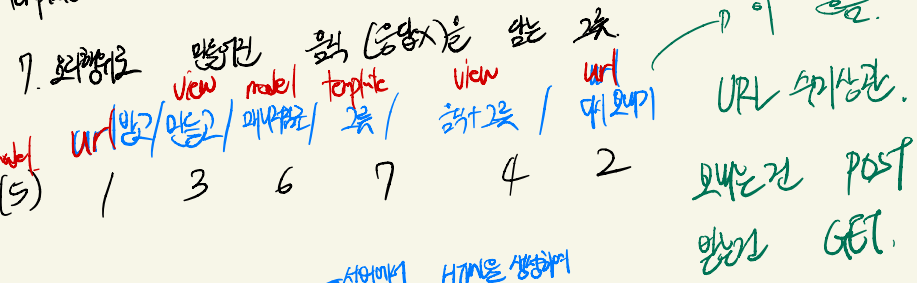
view는 요청엔

3. 컨트롤러가 받은 요청로 새로운 view 생성
4. 서비스에 보낸 요청(주소 + id)은 밑에서 view 생성

model은 낱고 매서드

5. view 객체를 특정한 객체로 작성하는 낱고 매서드.
6. view 객체를 저장하거나 가져올 수 있는 낱고 매서드.

template는 글



- render : html 파일만 불러오기. (URL 변경 X.)
- redirect : 다른 URL로 보내기. (클라이언트 요청을 통해 서버로 이동함.)

Read
 ① URL path " / " , default 121.00 ~
 (views.home, name="home")

② View → views에 home 할지 path에 home.

③ model에 query (query, query)

④ template response 를 보임.

① URL 에 맞는지 |
 router (URL 번호 X.)

⑤ 응답 값 (query 결과) (class) (template)

④ template (html)

Create

① URL create

→ ② views.create

→ ③ 새 객체 보낼 정보, (title, content)

⑤ url views detail + new_blog_id
 ↳ path blog/ <id=blog-id>

⑥ redirect url 새로 URL

→ url views.detail → 링크가 없음.

Update

① URL update 이즈

↳ ② views.update
 ↳ views.edit.

(기존에 있던 url) (update query)

→ ③ views.update

④ model

⑤ template, (detail)

↳ url views.detail

기존에 있던 url 새로운 url 정보.

↳ redirect ('detail', old_blog_id)

pk=blog-id 에 맞는 동일한 객체인지 확인 (이름만 있음)

↳ url detail 새로 URL 번호 (링크가 있음)

Delete

① URL

delete/<int:'blog-id'>

↳ ② views.delete

→ ③

blog_id 파라미터 없.

↳ url이 없.

delete 실패 (404)

redirect home return

↳ url home url → 링크가 있음.