

## route

- / main (Get)
- / home (Get)
- / login (Get, Post)
- / singup (Get, Post)
- / mypage / :id (Get)
- / mypage / :id / load.
- / mypage / :id / save .
- / mypage / :id / scrap
- / mypage / :id / document
- / edit / :id → document id. (Get, Post)
- / edit / new . (Get, Post)
- / Search (Get, Post)
- / my document . → api
- / Select .
- / Analysis
- / Analysis / :id . → document id .
- / Analysis Execute . → api .

## 상세처리

### ① Main.

1. 상단으로 클릭시 "/" 가동됨.
2. "함께하기" 클릭시 "%Signup" 가동됨.
3. "이미회원가입가요" 클릭시
  - [ 세션정보 : "/home" 가동 ]
  - [ 세션정보X : "/login" 가동 ]

세션유지시간?

### ②-1 login

성공시 "/home" 으로 이동됨.

1. /login에서 Post 요청 email, pw 보내고 passport local에서 처리. login 버튼의 input type="submit"

2. google 로그인
  - [ app → github 처리 ]
  - [ github → app ]
3. kakao 로그인 "

local과  
마찬가지로  
세션정보 필요.

## ②-2 Sign up.

1. local 가입하면 postJoin → postLogin . ↗  
(가입하기 클릭) (nextc) home으로 리다이렉션.
2. google. ②-1과 같은
3. kakao ②-(와 같은).

## ③ Home.

자동으로 ?

1. 3고글의 "Home" 라우팅.
2. logout 클릭시. passport req.logout() 처리하고 세션정보 날림  
"/"로 리다이렉션.
3. mypage 클릭시. "/mypage/:id" 라우팅.
4. 웹툰 시작하기 클릭시 "/Search"로 라우팅
5. 아랫툰 시작하기 클릭시 "/~~analysis~~"로 라우팅.  
select

→ global 미들웨어,  
- passport 2가지로 사용한다

④ mypage/:id

이거랑 같은 맥락

클릭하면 Ajax랑 Ajax 라우팅이랑  
분리해서 적용할 것.

0. 모든 버튼 누르기 전에 컨텐츠 내의 element를 비워주어야 함.

1. 초기 로딩하자마자 fetch ('/mypage/:id/load')

Ajax 요청. (window.addEventListener('DOMContentLoaded',  
function () { fetch(...) }) ---

• /mypage/:id/load 라우트에서는 :id 추출 후 DB에서 찾은 후

then으로 넘겨줌. then 안에서 UI 셋팅을 적용하면 됨.

• save 누르면 fetch 인자로 value 값을 전달하고

fetch ('/mypage/:id/save') 라우트. 함수 분리해야  
함든 Ajax UI.

2. 프로필 수정 누기도 1. 과 같음

3. 험객자소서 스크랩 누르면 fetch ('/mypage/:id/scrap', {})

route에서 id (req.params) 가져온 후 DB에서 가져온다.

then으로 정복전달. ~~클릭해서~~ 배열로 받은 후 Ajax UI 적용.

→ 험객자소서 어떻게 가져올 건지?

4. 나의 이력서 누르면 ~~fetch ("my/page/:id/document")~~.

~~컨트롤러에서 "이디" 사용자 ID 아래로 자바스크립트로 DB에서~~

~~불어와야 함. 그리고 Ajax UI 구현할 때 a href~~

~~href는 DB에서 가져온 document의 ID를 넣어주어야 함.~~

/my/document을 Ajax 요청

⑤ edit/:id.

1. edit/:id로 라우팅. ID는 DB에서 정보 가져오고

view는 정보를 전달. ↴ 질문, 답변 Array

2. post 요청시 (등록버튼클릭) DB에 추가된 정보 합성.

3. 추후 버튼누르면 /my/page/:id로 리다이렉트 (req.user)

⑥ edit/hew.

1. Ⓛ 클릭시 /edit/hew GET 요청.

2. /edit/hew에서 post 요청 (등록 버튼클릭) ↳ 1번  
DB에서 Create element.

3. 추후 누르면 /my/page redirect.

## ⑦ Search.

1. 둘보기 누르면 /Search?O=D & ... query 넣을 때

페이지 리로드

2. query 바탕으로 (hotPost, GET 요청). DB에서

정보 조회 후 출력. mydocument.

3. 내자소서와 비교 클릭시 /search/compare

Ajax 요청, 라우트쪽에서 req.user 바탕으로 이전 설정과 DB에서  
만나다음 then으로 전환.

## ⑧ /Select.

1. 대상 작성하기 클릭시 /analysis/:id 라우팅.

2. 자소서 불어오기 클릭시 fetch('/mydocument')

Ajax 요청 여기에서 대자소서 리스트에서 갖고온  
href = "analysis/:id" 대입함.

## ① ~~/analysis~~ /analysis/:id.

1. 웹사이트에서 조회하거나 fetch (" /analysis/:execute")  
AJAX통신  
문제를 해결하는

2. 코드를 브라우저에서 DB XHR  
HTML Element의 value.  
저장 (새로운 스케마)  $\Rightarrow$  "실시간 분석" 결과값  
theh으로 받아오기.

## ② /analysis/:id.

구현도입.