

0. API 자동 문서화 확인

- HTTP 리퀘스트 요청 URL 경로 및 양식 & JSON양식 샘플 확인 가능

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/swagger>

https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/redo_c

1. 회원가입

- 회원가입 및 등록

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/accounts>

⇒ 회원가입에 필요한 정보들을 입력하면 인증 ACCESS TOKEN을 리턴해준다.

- POST JSON 요청 양식

```
{
  "username": "", <= 빈칸으로 이렇게 넘겨주어야 함
  "email": "email@naver.com", unique 데이터 설정으로 중복값이 입력되면 오류 반환
  "password1": "password1234#",
  "password2": "password1234#", <= 입력한 패스워드의 일치 여부 확인용
  "army_num": "21-12345678", <= unique 데이터 설정으로 중복값이 입력되면 오류 반환
  "army_rank": "corporal", <= 정해진 리스트에 있는 키워드가 입력되어야 함
  "name": "문제웅"
}
```

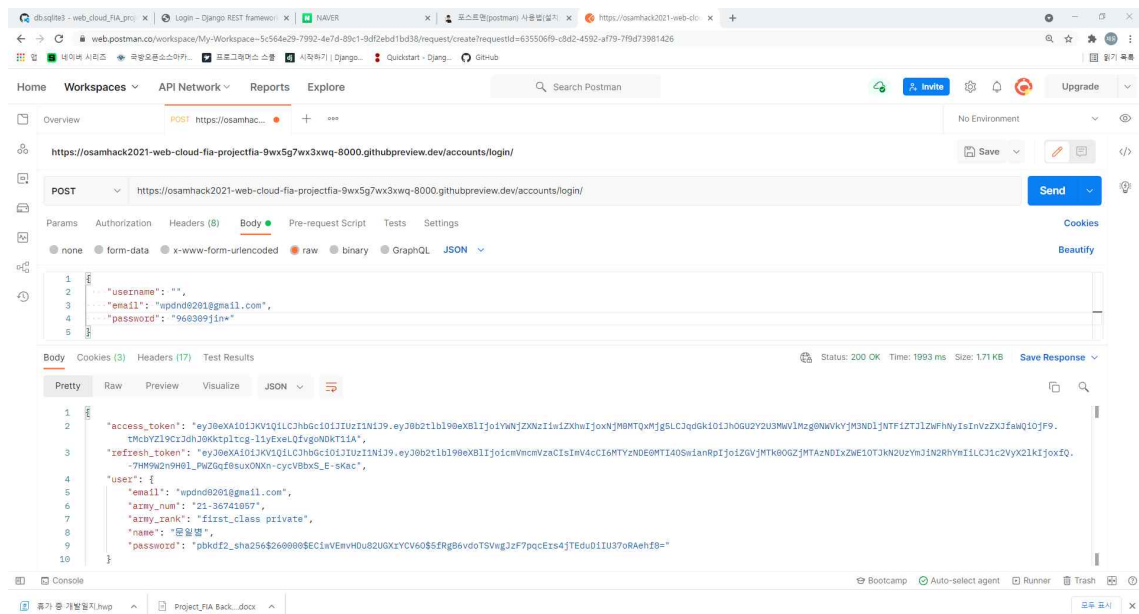
```
rank_lists = (
    ('private', '이병'),
    ('first_class private', '일병'),
    ('corporal', '상병'),
    ('sergeant', '병장'),
    ('executive', '간부 및 장교'),
)
```

2. 로그인

- 등록이 되어 있는 회원의 email과 password를 입력받아 POST하면 인증 ACCESS TOKEN을 리턴해준다. (주의사항 - 맨 뒤에 /를 잘 붙여주어야 된다.)

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/accounts/login/>

control + 마우스 휠로 확대하거나 이미지를 확대해서 보시면 잘 보이실 겁니다.



- POST JSON 요청 양식

```
{
  "username": "", <= 빈칸으로 넘겨 주어야 함
  "email": "email@naver.com", unique 데이터 설정으로 중복값이 입력되면 오류 반환
  "password": "password1234#" 가입 시 입력했던 패스워드 입력
}
```

3. 토큰 인증 방식 + HTTP Request에는 어떻게 요청해야 하는지

HTTP Request Header에

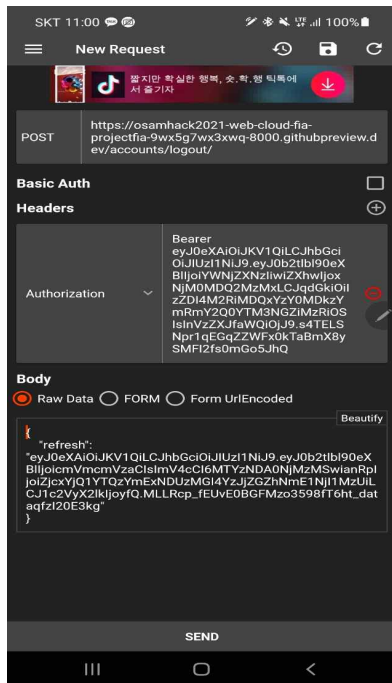
Authorization: Bearer YWxhZGRpbjpvucGVuc2VzYW1l

위 형식으로 항목을 추가해서 리퀘스트를 보내시면 됩니다. 향후 게시글 포스팅 및 수정 등 로그인 인증이 필요한 기능들에 대해서는 모든 리퀘스트에 이런식의 헤더와 로그인할 때 받았던 토큰값을 같이 보내주시면 됩니다.

4. 로그아웃

- JSON 양식으로 Refresh Token을 첨부하여 POST 요청하시면 헤더에 첨부되어 같이 전송된 ACCESS TOKEN이 폐기되고 해당 유저가 로그아웃 처리 됩니다.

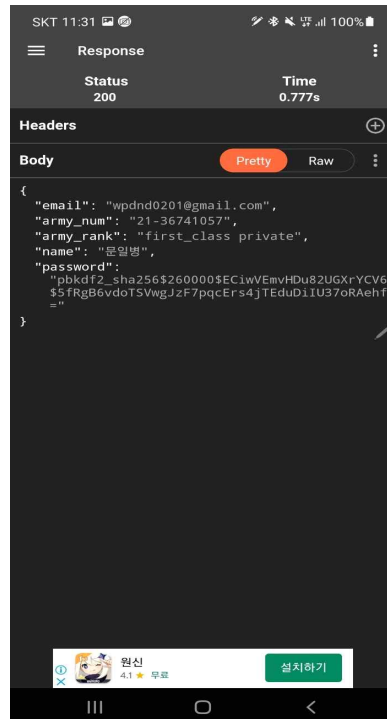
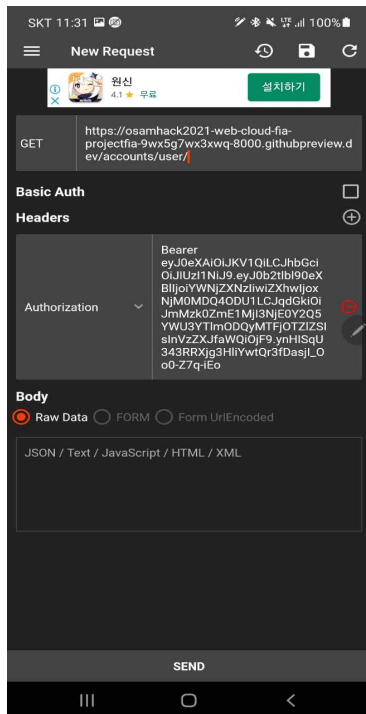
<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/accounts/logout/>



5. 사용자 정보

- 현재 로그인 되어 있는 사용자의 정보를 리턴해준다.(POST)

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/accounts/user/>

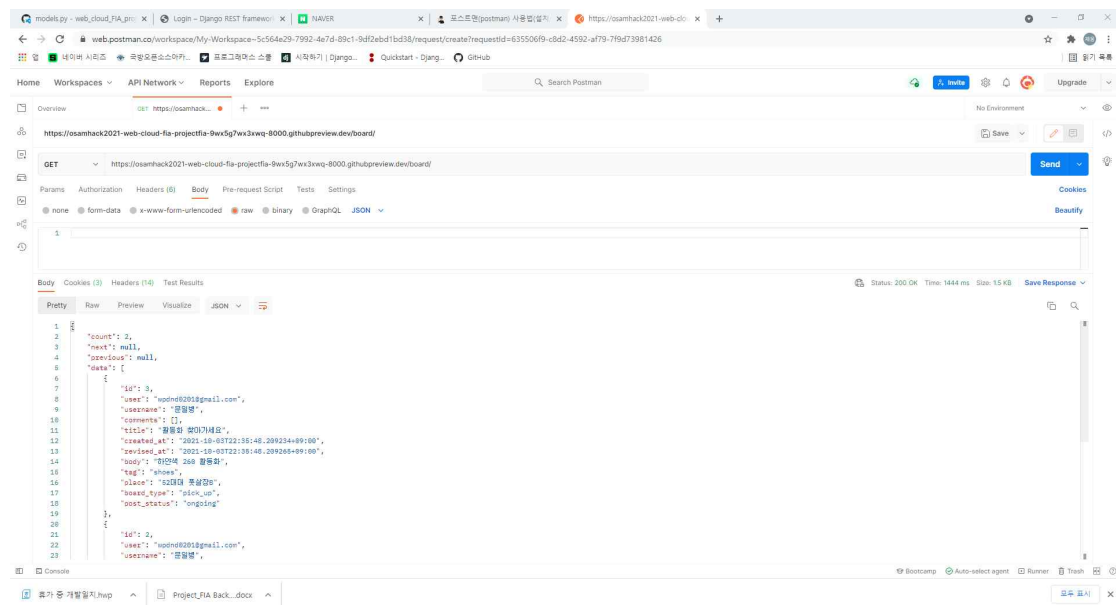


6. 게시물 전체 리스트 확인 (get)

- Project_FIA에 저장된 모든 게시글의 리스트를 전달받는다(한 번에 전달 받는 개수 조정 가능 => 설정값을 바꿔야 되니 추후 논의 필요)

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/>

(로그인 안한 비회원도 게시판에 접근 시 받아올 수 있는 정보입니다.)

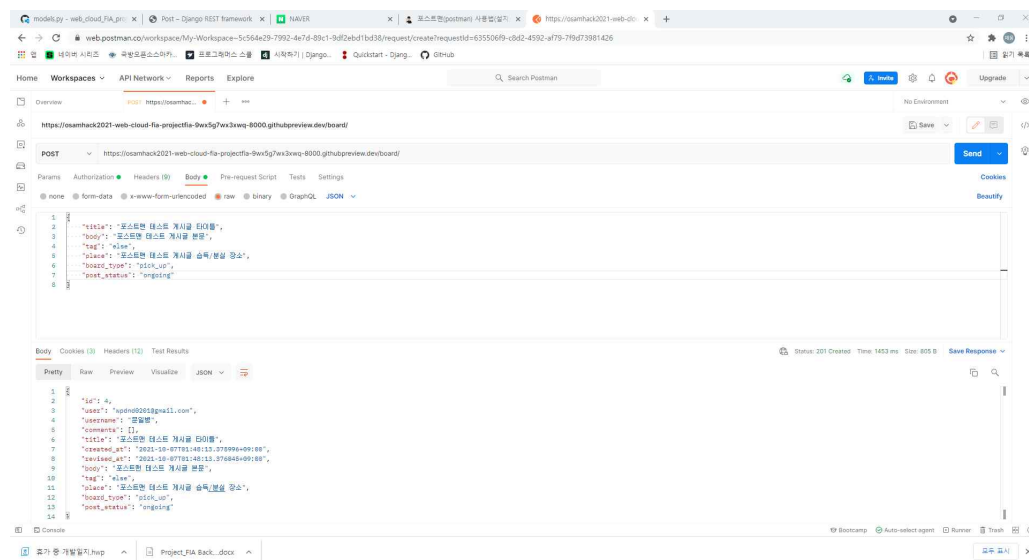


7. 게시물 포스팅 (POST)

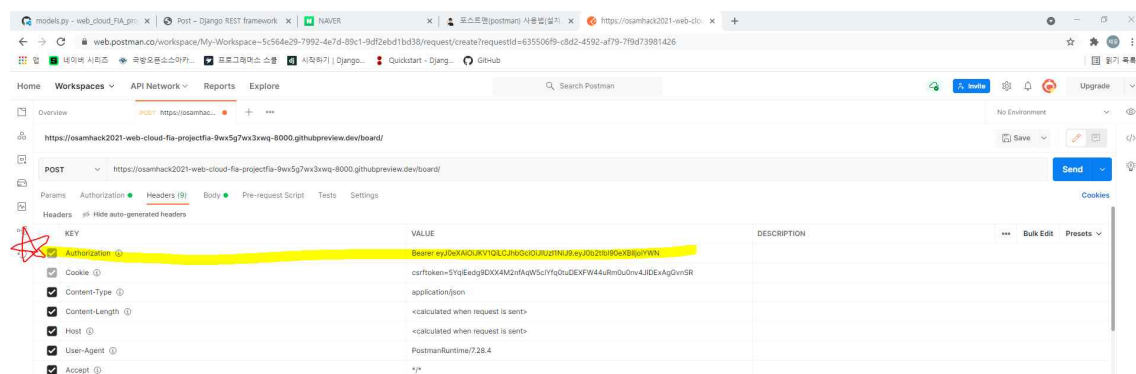
- 로그인된 유저의 토큰을 헤더에 첨부하고 게시글 JSON 양식을 함께 작성하여 게시글을 포스팅할 수 있다.

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/>

(게시판에 글 업로드 요청 결과)



(Authorization 헤더 양식)



- POST JSON 요청 양식

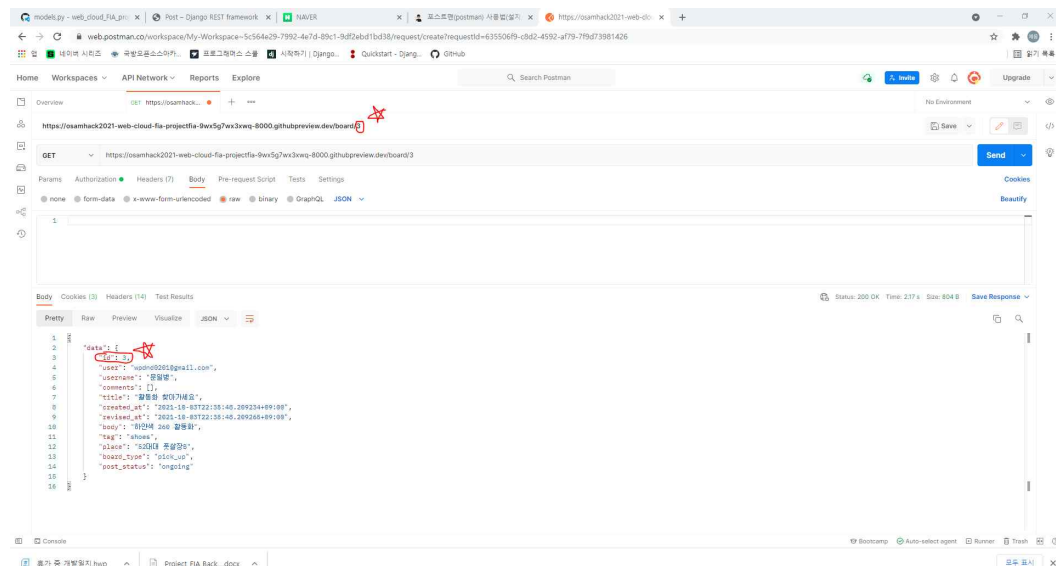
```
{
  "title": "",
  "body": "",
  "tag": null,
  "place": "",
  "board_type": null,
  "post_status": null
}
```

8. 게시물 세부 조회 => 단일 게시물 조회 및 수정기능 (GET, PUT)

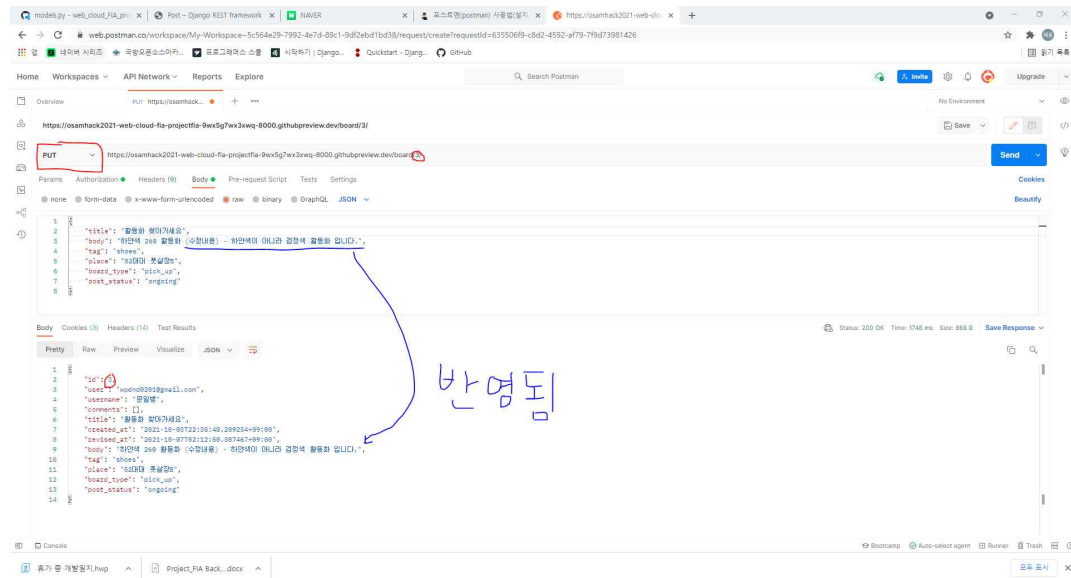
<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/4/>

- 게시글을 수정하기 위해서는 세부 조회 URL로 PUT 리퀘스트를 해야한다. 리퀘스트를 할 때 JSON 양식에 기존에 존재하던 게시글 양식을 불러와서 그대로 넣어주고 수정할 부분을 수정한 뒤 PUT 해서 업데이트 시켜준다.
- 세부 조회하고자 하는 게시글의 id 값을 url 뒤에 입력해서 GET 요청을 보낸다.(세부 조회)
- 수정하고자 하는 게시글의 id 값을 url 뒤에 입력해서 PUT 요청을 보낸다.(수정)
- 사용자의 게시글 수정 요청 뿐만 아니라 게시글 게시판 옮김 여부, 게시글 진행 상태 수정 및 변경 또한 이 요청으로 이루어져야 한다.
- 여기에서 수정 요청하는 인증 토큰의 주인 유저가 게시글을 작성한 유저와 일치해야 한다.
- 서버는 전송받은 인증 토큰과 유저의 정보가 다른 경우 접근 권한이 없다는 메시지를 준다.

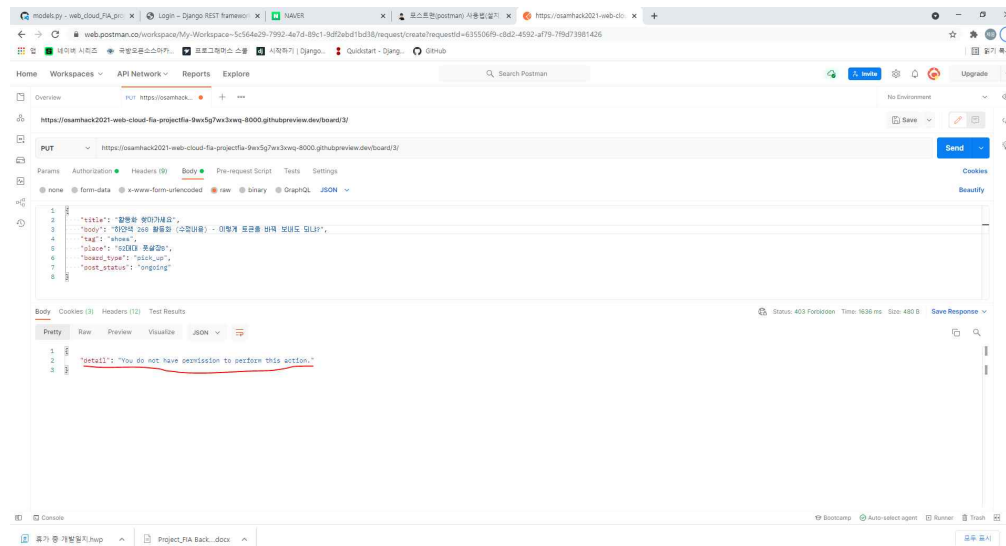
(게시글 세부 조회 get 요청)



(게시글 수정 요청)



(게시글 작성자가 아닌 다른 유저가 접근 시)



- 요청 JSON 양식 (※ 기존에 있던 데이터를 그대로 넣어주고 수정내용을 변경해야 함)

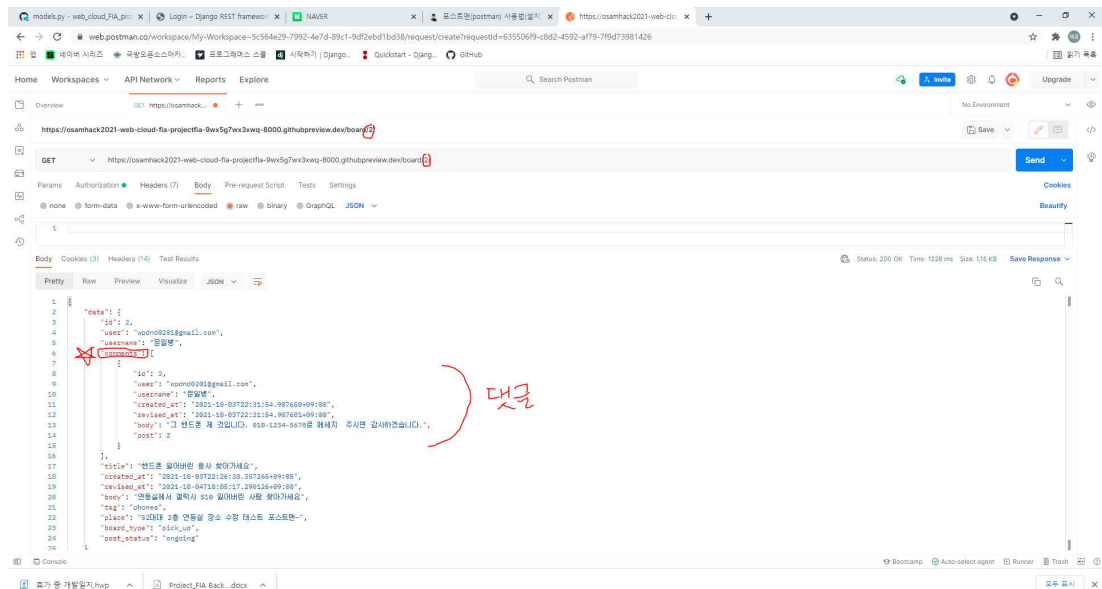
```
{
  "title": "활동화 찾아가세요",
  "body": "하얀색 260 활동화 (수정내용) - 하얀색이 아니라 검정색 활동화 입니다.", <=수정내용 추가
  "tag": "shoes",
  "place": "52대대 풋살장B",
  "board_type": "pick_up",
  "post_status": "ongoing"
}
```


}

9. 게시물에 대한 댓글 리스트 확인

- 게시물 세부 조회 후 받을 수 있는 JSON 결과 데이터에서 comments 컬럼에 접근하여 알아낼 수 있음.

(2번 게시글을 세부 조회한 후 거기에 달려 있는 댓글을 "comments" 항목을 통해 접근 가능할 수 있음)



10. 댓글 작성 (POST)

- 아래의 경로로 POST 요청을 하면 댓글을 추가할 수 있다.

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/comment/>

{

"body": "댓글 내용을 여기에 넣어주면 됩니다.", <= 댓글의 본문 저장

"post": 2 <= 댓글을 달고자 하는 게시글의 id (integer)

}

11. 댓글 세부 조회 => 수정기능 (PUT)

- 댓글의 경우 게시글의 id와는 독립적인 id들을 가지므로 게시글의 comments 컬럼을 통해 게시글에 달려 있는 댓글들의 id들을 파악하여 수정요청을 보내야 한다.
- 아래의 URL로 PUT 요청을 하면 해당 댓글의 내용을 수정할 수 있다.(id=3을 가진 댓글 수정)

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/comment/3>

```
{
  "body": "댓글 수정한 내용을 여기에 넣어주면 됩니다.", <= 댓글의 수정된 본문 저장
  "post": 2 <= 댓글을 달고자 하는 게시글의 id (integer)
}
```

12. 게시글 검색 기능 => 제목, 본문 일부분 검색 / 데이터베이스 필드 값 검색(다수 필드 & 동시검색 가능)

- 사용자가 게시글 전체 목록에서 원하는 내용 혹은 키워드를 포함하는 게시글들을 찾고 싶을 때 이용가능한 기능. 크게 서치엔진과 필터링 두가지 기능으로 나뉘며 한가지 방법만 사용할 수도 있고 두가지 방식을 &로 묶어서 모두 병행 가능하다.
- 게시글들에 대한 검색이 이루어지므로 board url에서 쿼리를 날린다.
- 쿼리 형식은 기본적인 board url에서 ?를 붙이고 search=~ 혹은 필터링하고자 하는 게시글 데이터의 컬럼명=(검색하고자하는 내용) 형식으로 사용 가능하다.

검색시 & 기능 사용가능, 정확하게 매칭이 되어야 검색 결과가 나온다.

`filter_fields = ['user__email', 'tag', 'board_type', 'post_status']`

검색시 search=... 형태로 사능 가능, 검색 내용이 포함된 결과물들을 보여준다.

`search_fields = ['title', 'body',]`

12-1. Search 엔진만 사용하는 경우

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?search=핸드폰>

=> 전체 게시글 리스트에서 제목, 본문에 핸드폰이라는 단어가 들어가는 게시글들의 리스트를 JSON 형식 데이터로 반환해준다.

12-2. 단일 데이터 컬럼에 대해 Filtering만 사용하는 경우

<https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?tag=phones>

혹은

https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?user_email=&tag=phones&board_type=&post_status=

=> 게시글의 tag 컬럼의 데이터가 phones인 게시글들의 리스트를 반환해준다.

12-3. 복수 데이터 컬럼에 대해 Filtering만 사용하는 경우

- 위의 filter_fields에 명시된 컬럼들을 조합하여 필터링이 가능하다.

https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?user_email=wpdnd0201@gmail.com&tag=shoes

혹은

https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?user_email=wpdnd0201@gmail.com&tag=shoes&board_type=&post_status=

=> 게시글 작성자의 이메일이 wpdnd0201@gmail.com 이고 태그가 shoes인 게시글 리스트를 보여준다.

12-4. Search와 Filtering 조합 사용 => & 활용

https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?user_email=wpdnd0201@gmail.com&search=활동화

혹은

https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?user_email=wpdnd0201@gmail.com&tag=&board_type=&post_status=&search

=> 게시글의 작성자 이메일이 wpdnd0201@gmail.com 이면서 게시글의 본문이나 제목에 활동화 단어가 포함이 된 게시글 리스트를 보여준다.

- 위에 두가지 url 중 사용하시기 더 편한 url을 선택해서 사용하시면 될 듯 합니다.

13. 로그인 중인 사용자가 작성한 글만 따로 요청하는 방법

- 게시글 전체 리스트를 보여주는 url에서 ?를 붙이고 user_email=(검색하고자 하는 이메일)을 입력해서 get요청을 하면 된다.

https://osamhack2021-web-cloud-fia-projectfia-9wx5g7wx3xwq-8000.githubpreview.dev/board/?user_email=wpdnd0201@gmail.com

Pagination 사용법

- 추후 데모 데이터를 많이 넣고 리스트를 반환받게 되면 10개씩 끊어서 보내주게끔 설정을 해두었습니다. 게시글 리스트를 받는 양식을 보시면 아래처럼 나오게 될 겁니다.

"count": 4, => 전체 게시물 수

"next": "http://localhost/board/?page=4", => 다음 페이지 주소

"previous": "http://localhost/board/?page=2", => 이전페이지 주소

"data": => JSON 리턴 데이터

※ 추후 한 번에 넘겨받는 게시글 리스트 개수를 변경하고자 하신다면 말씀해주시면 제가 원하시는 개수로 수정해드리겠습니다.