

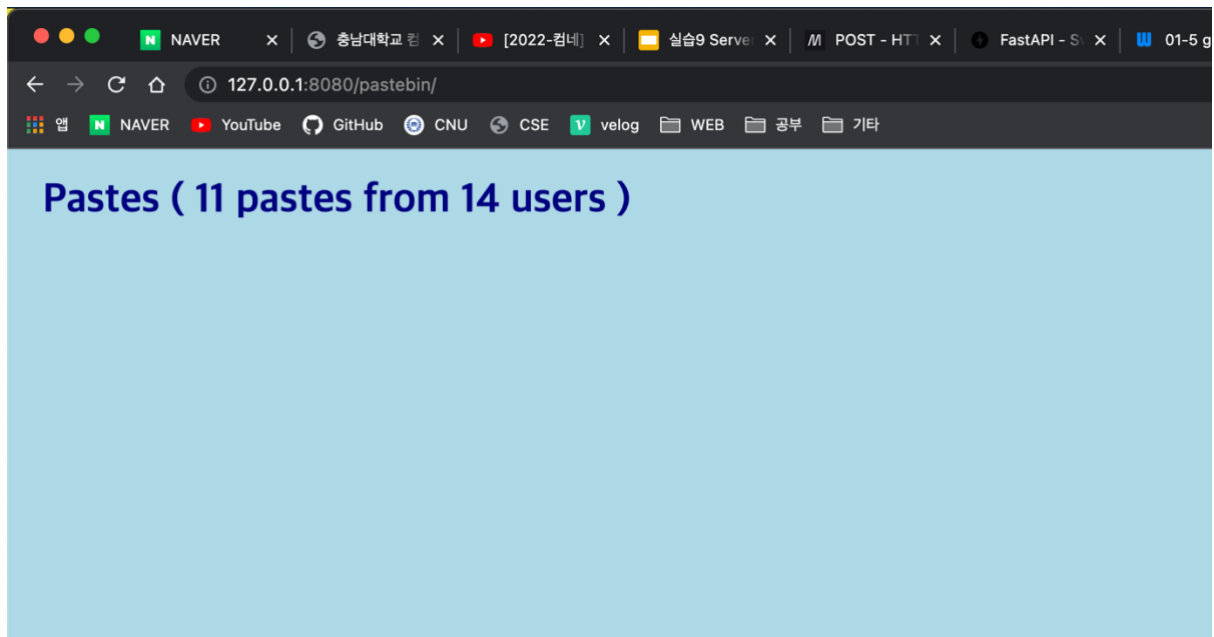
컴퓨터 네트워크

[실습 9]

컴퓨터공학과

201702042 우정균

1. Index

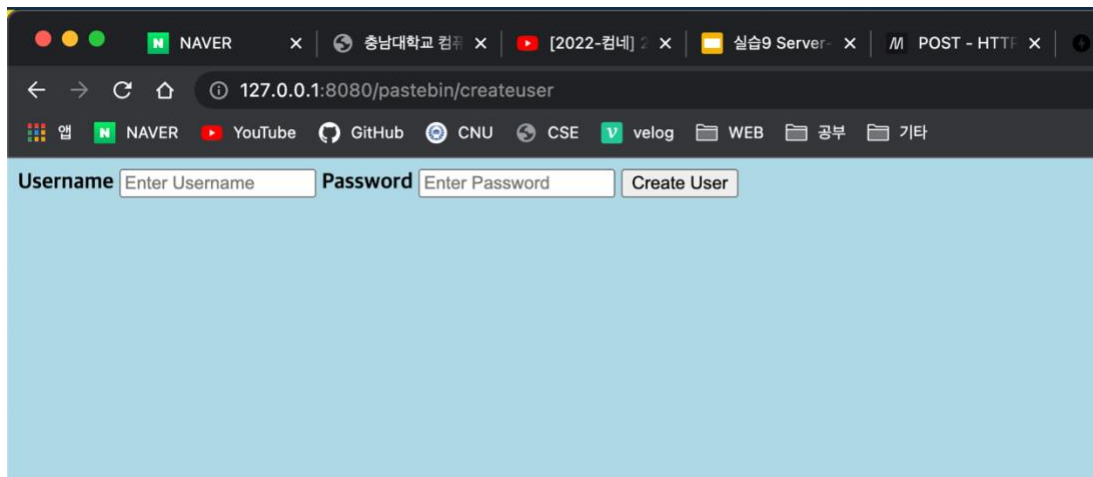


Index 페이지

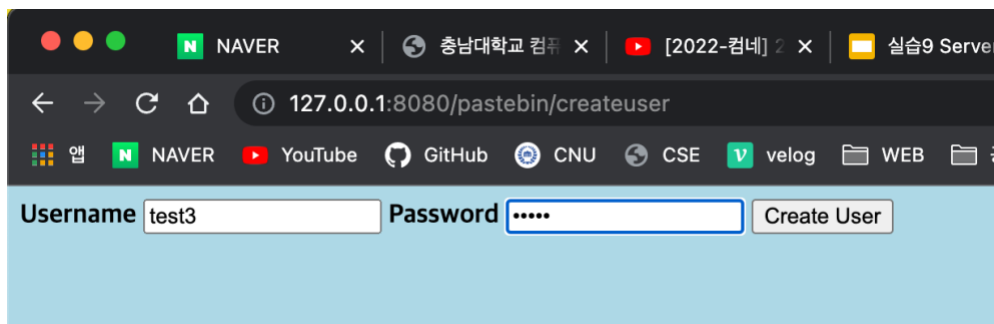
static 폴더의 css 파일도 잘 연동이 된다.

코드는 기본 제공 코드를 복사 붙여넣기 했으므로 생략

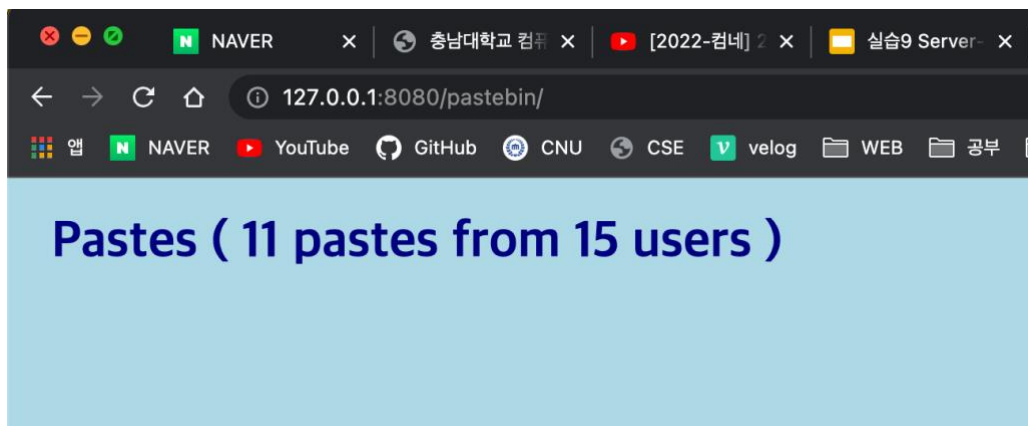
2. create user



/createuser url 에 접속한 페이지이다.



유저 이름과 패스워드를 입력 후 버튼을 누르면



Index 페이지로 이동하고, 새로 추가된 유저 수가 반영이 된다.

```

@bp.route(f'/createuser', methods=['GET', 'POST'])
def create_user():
    if request.method == 'POST':
        url = f'{endpoint}/users/'
        data = {'username': request.form['username'],
                'password': request.form['password']}
        headers = {'Accept': 'application/json',
                   'Content-Type': 'application/json;charset=utf-8'}
        method = 'POST'
        req = urllib.request.Request(url=url,
                                     data=json.dumps(data).encode('utf-8'),
                                     headers=headers,
                                     method=method)
        urllib.request.urlopen(req)
        return redirect('/pastebin/')
    elif request.method == 'GET':
        return render_template('createuser.html')

```

create_user() 함수의 소스코드

웹 페이지를 로드하기 위한 GET 메소드와

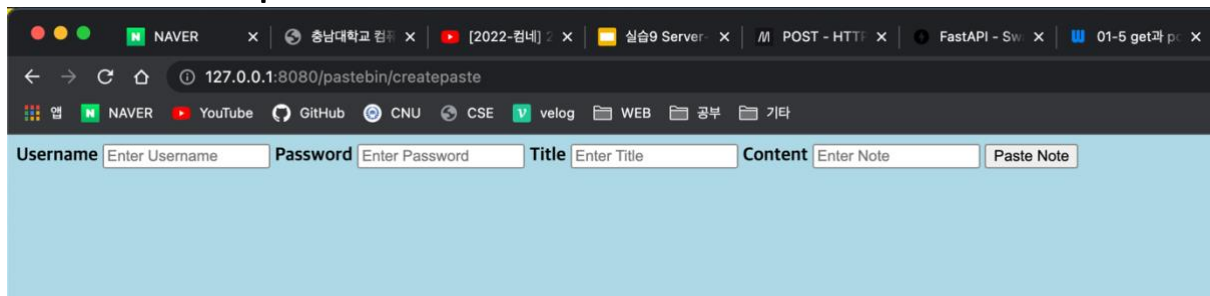
유저를 저장하는 요청의 POST 메소드를 각각 분기처리해서 진행한다.

POST 요청일 경우 /users/ url 로 username 과 password 를 data 에 담아서 POST 요청을 한다.

요청이 완료되면 index url 로 리다이렉션한다.

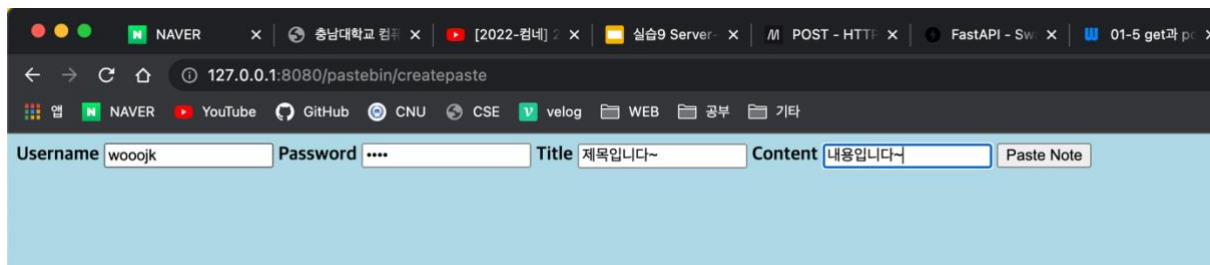
GET 요청일 경우 createuser.html 파일을 렌더링한다.

3. create paste



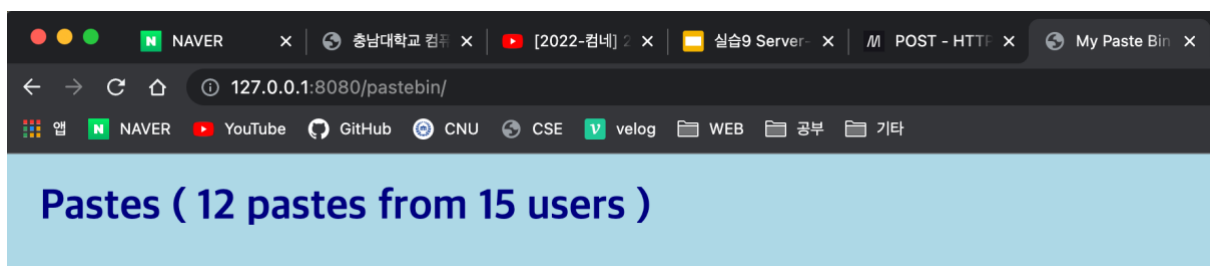
A screenshot of a web browser showing the 'create paste' form. The browser's address bar displays '127.0.0.1:8080/pastebin/createpaste'. The form includes input fields for 'Username' (placeholder: 'Enter Username'), 'Password' (placeholder: 'Enter Password'), 'Title' (placeholder: 'Enter Title'), 'Content' (placeholder: 'Enter Note'), and a 'Paste Note' button. The browser's tab bar shows several open tabs, including 'NAVER', '충남대학교 컴', '[2022-컴네]', '실습9 Server', 'M POST - HTTP', 'FastAPI - Sw', and '01-5 get과 p'.

create paste 페이지에 접속한 모습



A screenshot of the 'create paste' form with the following fields filled out: 'Username' is 'woojk', 'Password' is masked with '****', 'Title' is '제목입니다~', and 'Content' is '내용입니다~'. The 'Paste Note' button is visible to the right of the content field. The browser's address bar and tab bar are the same as in the previous screenshot.

username 과 password 를 입력하고, title 과 content 를 채워서 버튼을 누르면



A screenshot of the 'index' page of the web application. The browser's address bar shows '127.0.0.1:8080/pastebin/'. The page displays the text 'Pastes (12 pastes from 15 users)' in a large, bold, blue font. The browser's tab bar shows the same tabs as before, with an additional 'My Paste Bin' tab added.

index 페이지로 이동하며 pastes 개수가 업데이트 된다.

```

@bp.route(f'/createpaste', methods=['GET', 'POST'])
def create_paste():
    if request.method == 'POST':
        url = f'{endpoint}/users/{request.form["username"]}/verify/?password={request.form["password"]}'
        data = None
        headers = {'Accept': 'application/json'}
        method = 'GET'
        req = urllib.request.Request(url=url,
                                     data=json.dumps(data).encode('utf-8'),
                                     headers=headers,
                                     method=method)
        try:
            res = urllib.request.urlopen(req)
            if(res.getcode() == 200) :
                url = f'{endpoint}/users/{request.form["username"]}/pastes/?password={request.form["password"]}'
                data = {'title': request.form['title'],
                        'content': request.form['content']}
                headers = {'Accept': 'application/json',
                           'Content-Type': 'application/json;charset=utf-8'}
                method = 'POST'
                req = urllib.request.Request(url=url,
                                             data=json.dumps(data).encode('utf-8'),
                                             headers=headers,
                                             method=method)
                urllib.request.urlopen(req)
                return redirect('/pastebin/')
            except urllib.error.HTTPError:
                return redirect('/pastebin/createpaste')
        elif request.method == 'GET':
            return render_template('createpaste.html')

```

create paste 의 소스코드

create user 와 마찬가지로 페이지를 로드할 GET 요청과 paste 저장을 요청하는 POST 로 나뉘어져 있다.

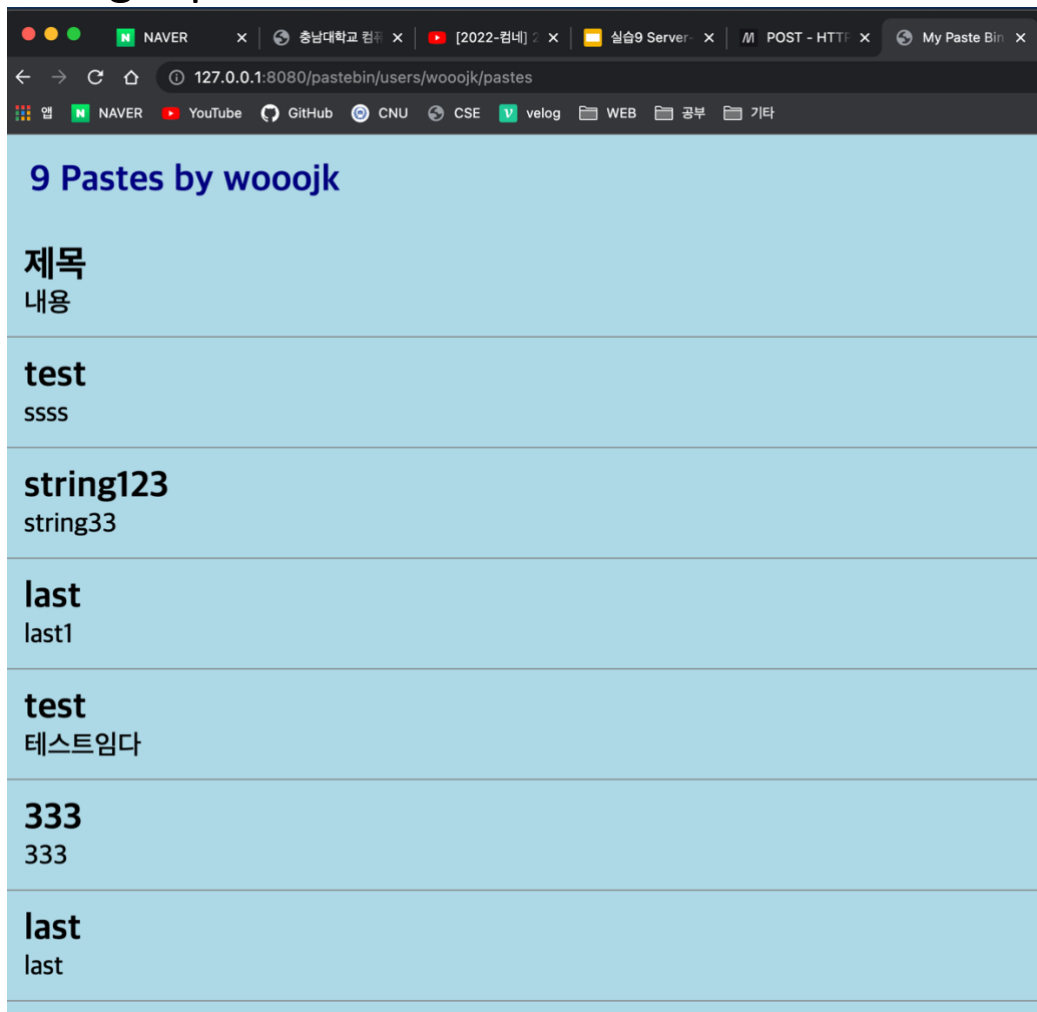
create user 와 다른 점은, POST 요청이 들어올 경우,
먼저 /users/{username}/verify/?password={password} url 에서
해당 name 과 password 가 올바른 값인지 확인을 한다.

저장된 username 과 password 가 아닐 경우 페이지를 새로고침한다.

올바른 username 과 password 일 경우 /users/{username}/pastes/?password url 로
pastes 저장 요청을 날린다.

요청이 끝나면 index 페이지로 리다이렉션한다.

4. get pastes



wooojk 유저의 paste 들을 가져온 모습의 페이지이다.

```
<!DOCTYPE html>
<html>
  <title>My Paste Bin - Pastes</title>

  <link rel="stylesheet" href="{{ url_for('mybp.static', filename='mystyle.css') }}" />

  <body>
    <h1>{{ count_pastes }} Pastes by {{ user_name }}</h1>
    {% for item in pastes %}
    <div id="pastes_container">
      <div id="title">{{item.title}}</div>
      <div id="content">{{item.content}}</div>
    </div>
    {% endfor %}
  </body>
</html>
```

pastes 페이지의 html 파일이다.

jinja2 의 문법을 사용해 pastes 의 데이터들을 html 로 받았고, css 로 스타일링 해줬다.

```

@bp.route(f'/users/<username>/pastes', methods=['GET'])
def get_pastes(username):
    count_pastes = 0
    url = f'{endpoint}/users/{username}/pastes/'
    data = None
    headers = {'Accept': 'application/json'}
    method = 'GET'
    req = urllib.request.Request(url=url,
                                data=data,
                                headers=headers,
                                method=method)
    with urllib.request.urlopen(req) as f:
        data = json.loads(f.read())
        count_pastes = len(data)

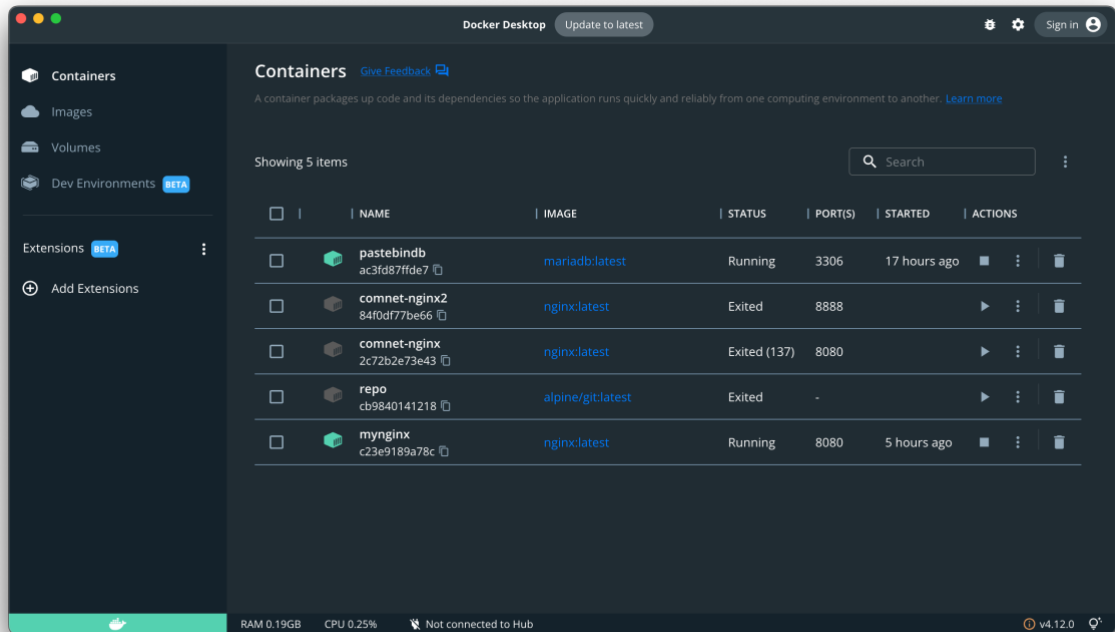
    return render_template('pastes.html',
                           user_name=username,
                           count_pastes=count_pastes,
                           pastes=data)

```

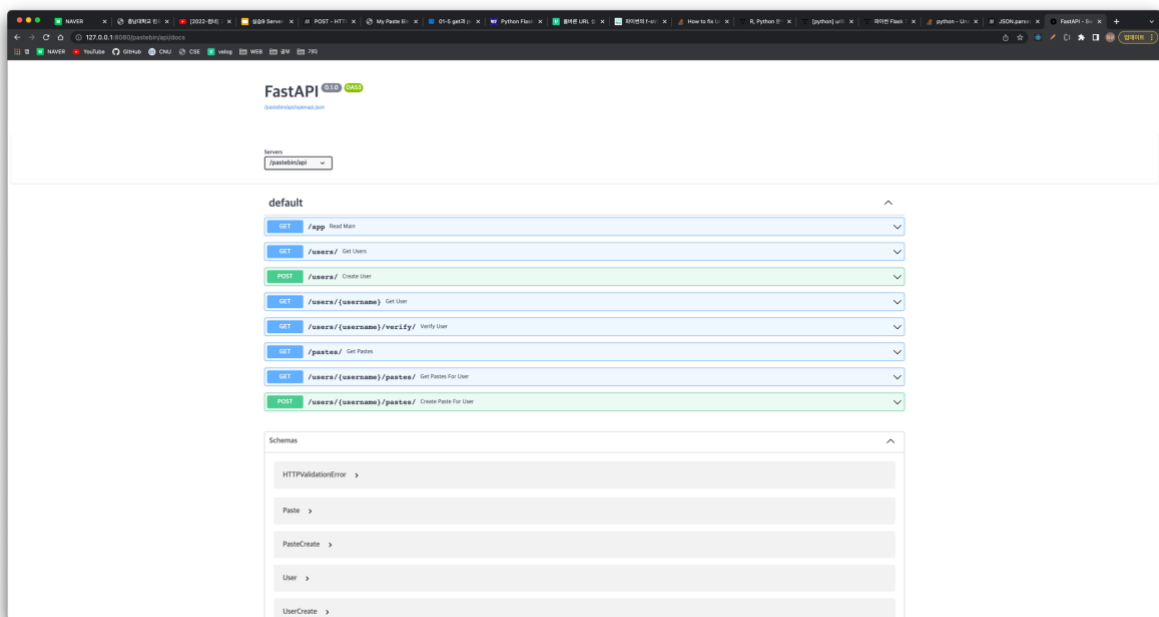
get pastes 소스코드이다.

/users/{username}/pastes/ url 에서 username 의 pastes 들을 가져오고,
username 과 pastes 개수, pastes 를 pastes.html 로 보내서 렌더링한다.

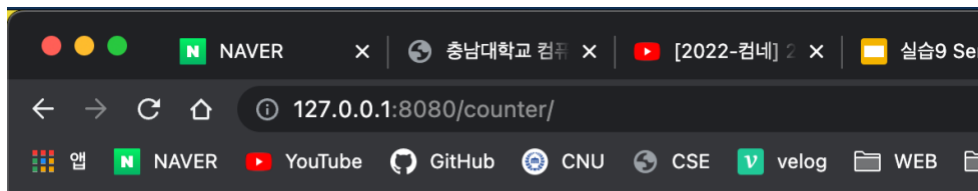
5. 그 외



docker 인증샷



:8080/pastebin/api/docs 로 들어가면 이전 과제에서 했던 fastAPI 의 문서가 나온다.



index.html

총 방문자: 82

:8080/counter 로 들어가면 방문자를 나타내는 페이지가 나온다.

```
server {  
    listen 8080;  
  
    location /counter/ {  
        proxy_pass http://host.docker.internal:8888;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header Host $host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    }  
  
    location /pastebin/api/ {  
        proxy_pass http://host.docker.internal:8889/;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header Host $host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    }  
  
    location /pastebin/ {  
        proxy_pass http://host.docker.internal:8890/pastebin/;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header Host $host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    }  
}
```

위 사진은 nginx.conf 파일의 location 을 설정한 모습이다.