

고급 웹프로그래밍 과제2

2020113486 김대건

1. 웹 서버 인프라 환경

배포환경 : AWS EC2

라이브러리 및 프레임워크 : TypeScript, NestJS, HandleBars(Template Engine)

2.

index.hbs 에 WAS 로부터 클라이언트의 접속시간에 대한 접속시간을 렌더링할 템플릿을 작성합니다

```
1  <html lang='ko'>
2    <head>
3      <meta charset='UTF-8' />
4      <meta name='viewport' content='width=device-width, initial-scale=1.0' />
5      <title>Document</title>
6
7      <link rel='stylesheet' href='/index.css' />
8    </head>
9
10   <body>
11     <div class='time'>접속시간 :
12       {{year}}년
13       {{month}}월
14       {{date}}일
15       {{hour}}시
16       {{minute}}분
17       {{second}}초
18     </div>
19
20     {{>nav}}
21
22     <div>
23       <h1>돼지고기 김치찜</h1>
24       <img src='/background.jpg' alt='' />
25     </div>
26   </body>
27 </html>
```

나머지 음식에 대한 기원과 레시피를 보여줄 템플릿을 작성합니다

```
1 <html lang='ko'>
2 <head>
3   <meta charset='UTF-8' />
4   <meta name='viewport' content='width=device-width, initial-scale=1.0' />
5   <title>Document</title>
6
7   <link rel='stylesheet' href='/index.css' />
8 </head>
9
10 <body>
11   {{>nav}}
12
13   <section>
14     <h3>김치점의 기원</h3>
15
16     <p>김치점의 역사는 상세히 알려져 있지 않다. 일반적으로 한국의 가정에서
17       흔하게 구할 수 있는 재료인 배추김치와 저렴한 고기나 생선을 이용하여 한
18       솥에 끓여내는 요리이기 때문에 일반 가정집에서 자연 발생한 요리라고
19       추측되고 있다.</p>
20
21   </section>
22
23   <section>
24     <h3>레시피</h3>
25
26     <p>1. 청주 3, 다진 마늘 1, 생강가루 0.3, 후춧가루 약간을 넣어 돼지고기
27       밀간을 해주세요. 김치점에 넣을 고기는 큼직하게 넣거나 또는 덩어리로
28       넣어주고 먹기 직전에 가위로 싹둑싹둑 잘라 드심 됩니다.</p>
29
30     <p>2. 김치 1/4포기를 준비해주세요. *오래된 묵은지는 흐르는 물에 김칫소를
31       씻어 사용하세요. 또 너무 신 김치는 미리 설탕을 넣어 조물조물 무쳐 주셔도
32       되고요. 그럼 신맛이 달아납니다.</p>
33
34     <p>3. 양파는 채 썰어 주고 대파, 청양고추는 어슷 썰어 주세요.</p>
35
36     <p>4. 물 3컵을 그릇에 담고 양념재료인 된장 0.5, 국간장 5, 설탕 2를 넣고
37       고루 섞어 주세요. 여기에 새우젓 1숟가락 정도 추가하셔도 좋아요.</p>
38
39     <p>5. 냄비에 밀간해 놓은 돼지고기를 넣고 김치도 함께 넣어주세요.</p>
40
41     <p>6. 그리고 양념해 놓은 육수를 모두 붓고</p>
42
43     <p>7. 고춧가루 3, 양파, 대파, 청양고추를 모두 넣고 1시간 30분 정도 푹
44       끓여주세요. 처음에는 뚜껑을 열고 센 불에서 10분 정도 끓여주다 (그래야지
45       고기 잡내가 날아가요) 그다음 뚜껑을 닫고 불은 약불로 줄이고 뽕글글하게
46       익혀줍니다. 김치가 가정마다 염도, 맵기도 그렇고 신맛도 모두 다르니 이럴
47       땐 설탕과 고춧가루를 조절해 가며 입맛에 맞게 간을 해주세요. 김치가 찢
48       때는 물에 살짝 씻어주거나 김칫소를 털어주거나 국물을 짜주세요. 1시간
49       이상 푹 끓이다 보면 고기에서 기름이 나와 김치에도 배어 윤기가 나고
50       고기는 아들아들하니 연해집니다.</p>
51
52     <p>8. 돼지고기 김치점 완성 ^^ 김치점 하실 땐 돼지고기에 밀간을 해주심
53       잡내를 없앨 수 있으니~기억해 주시고요. 무엇보다 신선한 고기 구입이
54       우선이겠쥬.
55   </p>
56 </section>
57 </body>
58 </html>
```

음식에 대한 별명을 붙이는 nickname.hbs 템플릿을 작성하고, 클라이언트 사이드 스크립팅을 할 index.js 를 작성합니다

```
1 document.querySelector('#btn').addEventListener('click', () => {
2   let nickname = document.querySelector('#nickname').value;
3   console.log(nickname);
4   document.querySelector('#text').innerHTML = nickname;
5 });
6
```

NestJS 의 진입점 main.ts 에서 템플릿 엔진 설정을 하고,

```
1 import { NestFactory } from '@nestjs/core';
2 import { AppModule } from './app.module';
3 import { NestExpressApplication } from '@nestjs/platform-express';
4 import * as path from 'path';
5 import * as hbs from 'hbs';
6
7 async function bootstrap() {
8   const app = await NestFactory.create<NestExpressApplication>(AppModule);
9
10  app.useStaticAssets(path.join(__dirname, '..', 'public'));
11  app.setBaseViewsDir(path.join(__dirname, '..', 'views'));
12  app.setViewEngine('hbs');
13  hbs.registerPartials(path.join(__dirname, '..', 'views', 'partials'));
14
15  await app.listen(3000);
16 }
17 bootstrap();
18
```

Controller 에서 appService 를 주입받아 템플릿 엔진이 보여줄 데이터를 바인딩합니다.

```
1 import { Controller, Get, Render } from '@nestjs/common';
2 import { AppService } from './app.service';
3
4 @Controller()
5 export class AppController {
6   constructor(private readonly appService: AppService) {}
7
8   @Get()
9   @Render('index')
10  public renderHome() {
11    return this.appService.getAccessTime();
12  }
13
14  @Get('/detail')
15  @Render('detail')
16  public renderDetail() {}
17
18  @Get('/nickname')
19  @Render('nickname')
20  public renderNickName() {}
21 }
22
```

```

1  import { Injectable } from '@nestjs/common';
2
3  @Injectable()
4  export class AppService {
5    public getAccessTime() {
6      const date: Date = new Date();
7
8      return {
9        year: date.getFullYear(),
10       month: date.getMonth(),
11       date: date.getDate(),
12       day: date.getDay(),
13       hour: date.getHours(),
14       minute: date.getMinutes(),
15       second: date.getSeconds(),
16     };
17   }
18 }
19

```

3. 결과화면 (<http://52.78.192.45/>)





