





Midterm / 중간고사

UT-NodeJS Midterm Checklist / 중간고사 체크리스트

TEST: Instructions / 시험: 지시사항

Unit 2 Capstone Detailed Steps / 유닛 2 캡스톤 자세한 단계

- (1) `npm init` 명령을 실행하여 Node.js 프로젝트를 시작하세요.
- (2) 프로젝트에 다음 패키지를 설치하세요.
 - `http-status-codes`
 - `express`
 - `ejs`
 - `express-ejs-layouts`
- (3) `main.js` (또는 `app.js`) 파일을 만들고 Express 서버를 설정하세요.
- (4) `views` 폴더 안에 `partials` 폴더를 만들고 다음 `partials`를 만드세요.
 - `header.ejs` - 모든 페이지에 사용되는 헤더
 - `footer.ejs` - 모든 페이지에 사용되는 푸터
 - `navigation.ejs` - 모든 페이지에 사용되는 네비게이션 바
 - `confetti.ejs` - thanks 페이지에 사용되는 confetti
- (5) 주어진 HTML 파일의 확장자를 `.ejs`로 변경하고 `views` 폴더에 `layout.ejs` 파일을 추가하여 레이아웃을 설정합니다.
 - `layout.ejs` - 모든 페이지의 레이아웃
 - `head` - `meta` 태그와 `title` 태그를 자신의 정보로 업데이트합니다.
 - `index.ejs` - GET 메소드로 접근 가능
 - `transportation.ejs` - GET 메소드로 접근 가능
 - `contact.ejs` - GET과 POST 메소드로 접근 가능
 - `thanks.ejs` - POST 메소드 후에 접근 가능
 - `error.ejs` - 에러 발생 시 접근
- (6) 다음 컨트롤러를 만들어 라우트와 예러를 처리하세요.
 - `homeController` - 애플리케이션의 모든 라우트(GET과 POST)를 처리하는 컨트롤러
 - `errorController` - 애플리케이션 예러를 처리하는 컨트롤러



01

셋업

npm init
npm install
파일 이름 바꾸기

npm init

Node.js 프로젝트 초기화

package name:
version:
description:
entry point: (main.js)
test command:
git repository:
keywords:
author: (이름)
license:

midterm-test-soln

- > pdfs
- > public
- > views
- ❖ .gitignore
- ❖ bonus.md
- ❖ checklist.test.md
- { package.json }**
- ❖ readme.en.md
- ❶ readme.md
- ❖ study-guide.md

```
o (base) PS C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (midterm-test-soln) midterm-test
version: (1.0.0)
● description: Midterm Test for Aaron
entry point: (index.js) main.js
test command:
git repository: (https://github.com/ut-nodejs/midterm-test.git)
keywords:
author: Aaron Snowberger
license: (ISC)
About to write to C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln\package.json:
과제 > solutions > midterm-test-soln > { package.json ...
```

```
1  {
2    "name": "midterm-test",
3    "version": "1.0.0",
4    "description": "Midterm Test for Aaron",
5    "main": "main.js",
6    "scripts": {
7      "test": "echo \\\"Error: no test specified\\\" && exit 1"
8    },
9    "repository": {
10      "type": "git",
11      "url": "git+https://github.com/ut-nodejs/midterm-test.git"
12    },
13    "author": "Aaron Snowberger",
14    "license": "ISC",
15    "bugs": {
16      "url": "https://github.com/ut-nodejs/midterm-test/issues"
17    },
18    "homepage": "https://github.com/ut-nodejs/midterm-test#readme"
19  }
20
```

npm install

패키지 설치

http-status-codes

(errorController.js)

express

(웹 앱 프레임워크)

ejs

(템플릿 엔진)

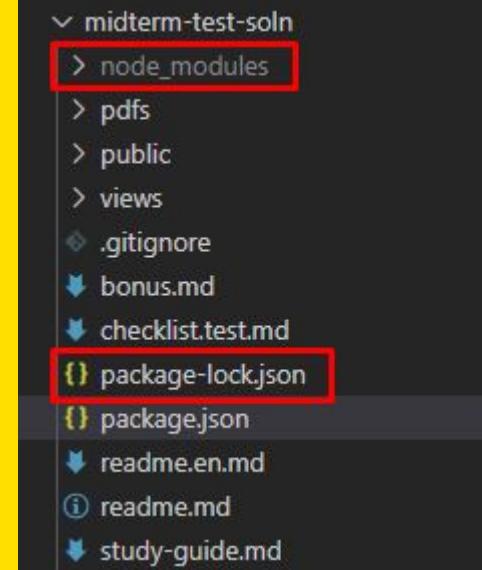
express-ejs-layouts

(layout.ejs 파일 사용)

```
(base) PS C:\Users\ADMIN\Desktop-soln> npm install http-status-codes express ejs express-ejs-layouts
added 75 packages, and audited 76 packages in 2s

9 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
○ (base) PS C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln>
```



파일 이름 바꾸기

html -> ejs

- contact.ejs
- error.ejs
- index.ejs
- thanks.ejs
- transportation.ejs

추가 파일 만들기

- index copy.ejs = **layout.ejs**
- /partials
 - header.ejs
 - navigation.ejs
 - footer.ejs

The screenshot shows three panels of a file explorer for a project named 'midterm-test-soln'. The left panel shows the initial state with HTML files: contact.html, error.html, index.html, thanks.html, and transportation.html. The middle panel shows the state after renaming, with files renamed to contact.ejs, error.ejs, index.ejs, thanks.ejs, and transportation.ejs. A red arrow points from each HTML file to its corresponding EJS file. The right panel shows the final state where all files have been renamed to EJS, and a new folder 'partials' has been created containing 'header.ejs', 'navigation.ejs', and 'footer.ejs'. A red box highlights the 'partials' folder, and a red arrow points from the 'index copy.ejs' file in the middle panel to the 'index.ejs' file in the right panel.

```
midterm-test-soln
  node_modules
  pdfs
  public
  views
    contact.html
    error.html
    index.html
    thanks.html
    transportation.html
    .gitignore
    bonus.md
    checklist.test.md
    package-lock.json
    package.json
    readme.en.md
    readme.md
    study-guide.md

midterm-test-soln
  node_modules
  pdfs
  public
  views
    contact.ejs
    error.ejs
    index.ejs
    thanks.ejs
    transportation.ejs
    .gitignore
    bonus.md
    checklist.test.md
    package-lock.json
    package.json
    readme.en.md
    readme.md
    study-guide.md

midterm-test-soln
  node_modules
  pdfs
  public
  views
    partials
      confetti.ejs
      footer.ejs
      header.ejs
      navigation.ejs
    contact.ejs
    error.ejs
    index.ejs
    layout.ejs
    thanks.ejs
    transportation.ejs
    .gitignore
    bonus.md
    checklist.test.md
    package-lock.json
    package.json
    readme.en.md
    readme.md
    study-guide.md
```

02

ejs 파일들

confetti.ejs (선택)
layout.ejs
/partials
페이지들

thanks.ejs에서 confetti.ejs를 만들 수 있다

줄 번호 옆에 있는 여는 HTML 태그에서
화살표를 클릭하여 코드를 접습니다.

The screenshot shows a code editor interface with the following details:

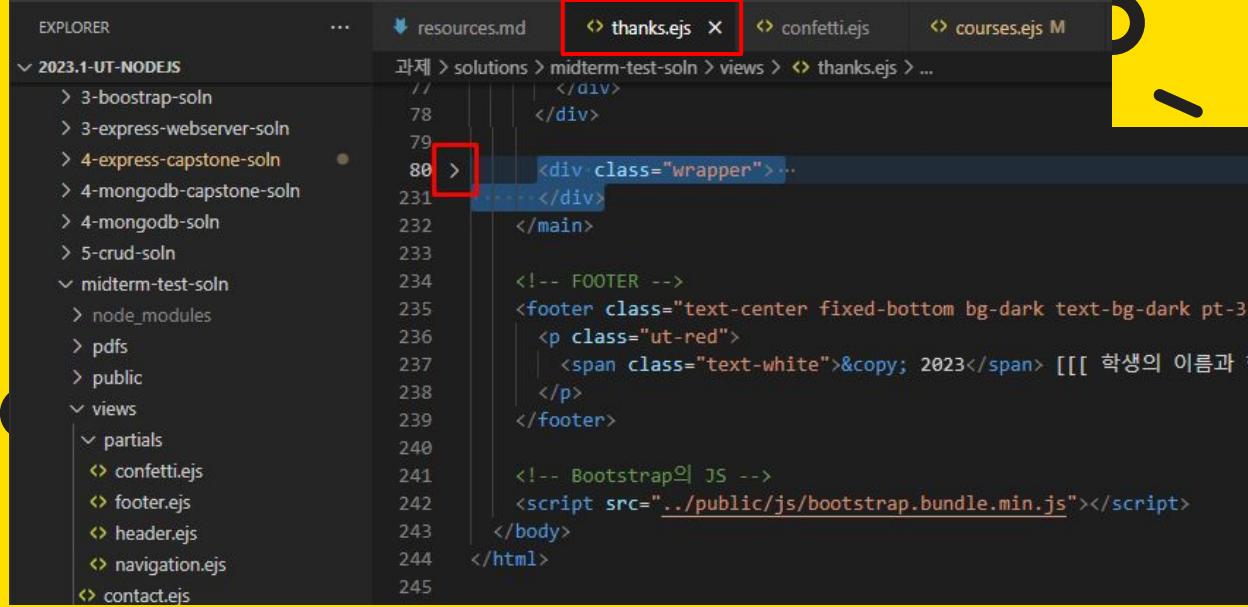
- EXPLORER** sidebar: Shows a tree view of the project structure under "2023.1-UT-NODEJS".
- CODE EDITOR**: The file "thanks.ejs" is open. The line of code being highlighted is: `

>

`.
- STATUS BAR**: Shows tabs for "resources.md", "thanks.ejs", "confetti.ejs", and "courses.ejs M".
- NAVIGATION**: Shows the file path: 과제 > solutions > midterm-test-soln > views > thanks.ejs > ...
- CODE**: The code consists of a series of nested div elements with specific class names ranging from "confetti-149" to "confetti-121".

thanks.ejs에서 confetti.ejs를 만들 수 있다

줄 번호 옆에 있는 여는 HTML 태그에서
화살표를 클릭하여 코드를 접습니다.



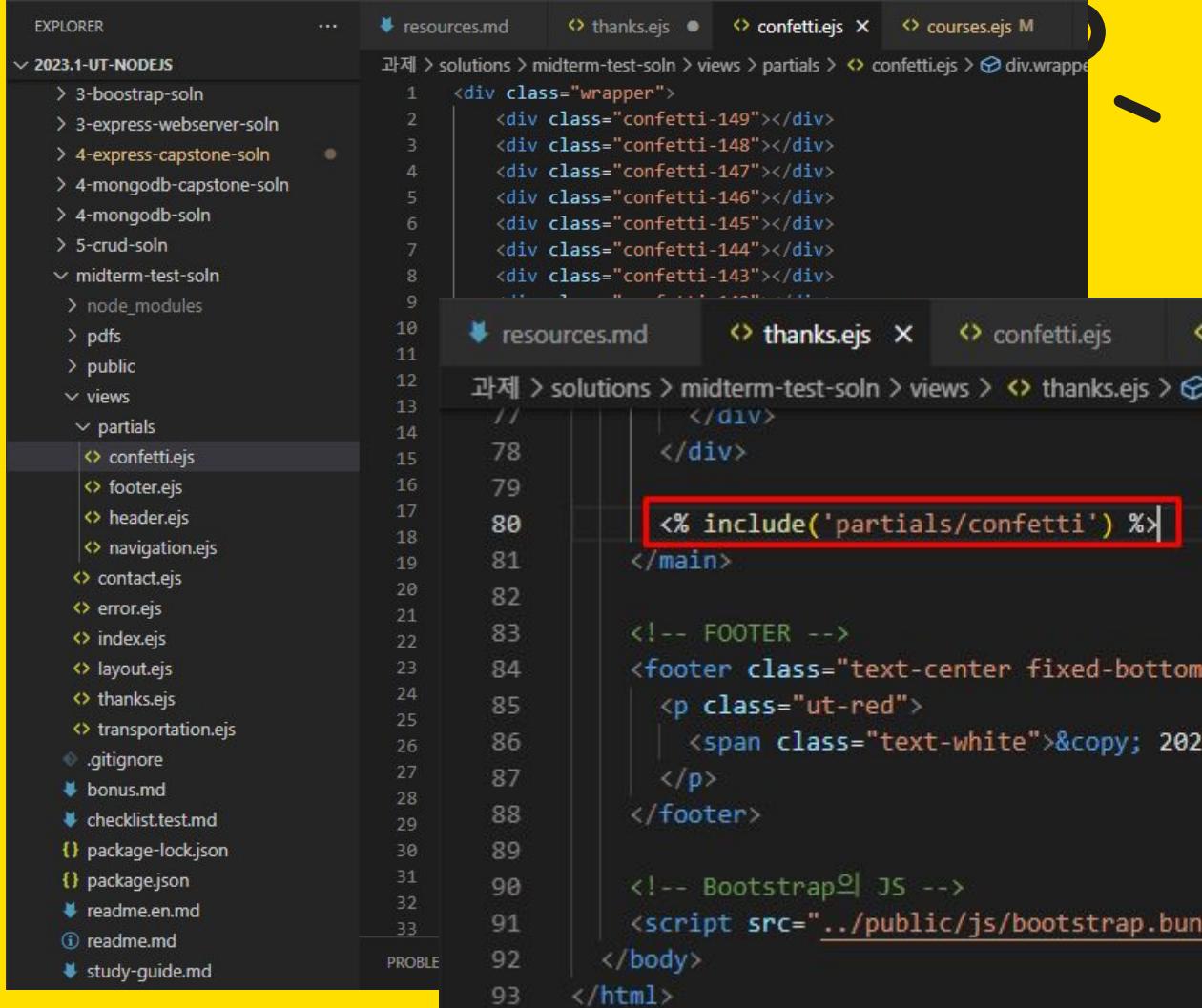
```
// resources.md
thanks.ejs X confetti.ejs courses.ejs M
...
Explorer ...
2023.1-UT-NODEJS
  > 3-bootstrap-soln
  > 3-express-webserver-soln
  > 4-express-capstone-soln
  > 4-mongodb-capstone-soln
  > 4-mongodb-soln
  > 5-crud-soln
  > midterm-test-soln
    > node_modules
    > pdfs
    > public
    > views
      > partials
        < confetti.ejs
        < footer.ejs
        < header.ejs
        < navigation.ejs
        < contact.ejs
...
줄 번호 옆에 있는 여는 HTML 태그에서
화살표를 클릭하여 코드를 접합니다.

과제 > solutions > midterm-test-soln > views > thanks.ejs > ...
80 > <div class="wrapper"> ...
231 </div>
...
</main>
...
<!-- FOOTER -->
<footer class="text-center fixed-bottom bg-dark text-bg-dark pt-3">
  <p class="ut-red">
    <span class="text-white">&copy; 2023</span> [[ [ 학생의 이름과
...
235
236
237
238
239
240
241
242
243
244
245
</p>
</footer>
<!-- Bootstrap의 JS -->
<script src="../public/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

confetti.ejs에서

confetti 부분만 붙여넣기

재사용 가능한 모든 코드를 별도의 파일에 복사하여 붙여넣습니다. 이렇게 하면 관리가 더 쉬워집니다. 그런 다음 하나의 파일만 모든 페이지 요소를 제어합니다. 코드를 변경하기 위해 모든 페이지를 수정할 필요는 없습니다. 해당 요소의 파일만 수정하면 됩니다.



```
EXPLORER ... resources.md <div class="wrapper">
  <div class="confetti-149"></div>
  <div class="confetti-148"></div>
  <div class="confetti-147"></div>
  <div class="confetti-146"></div>
  <div class="confetti-145"></div>
  <div class="confetti-144"></div>
  <div class="confetti-143"></div>
  ...
</div>

resources.md <div class="confetti-149"></div>
<div class="confetti-148"></div>
<div class="confetti-147"></div>
<div class="confetti-146"></div>
<div class="confetti-145"></div>
<div class="confetti-144"></div>
<div class="confetti-143"></div>
...
</div>

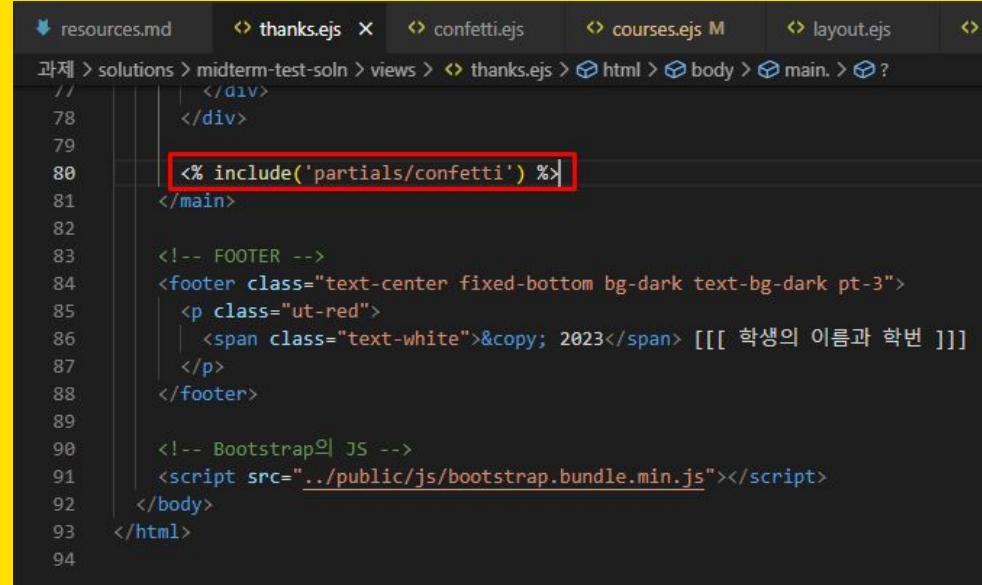
resources.md <div class="confetti-149"></div>
<div class="confetti-148"></div>
<div class="confetti-147"></div>
<div class="confetti-146"></div>
<div class="confetti-145"></div>
<div class="confetti-144"></div>
<div class="confetti-143"></div>
...
</div>
```

The code editor shows two tabs: 'confetti.ejs' and 'thanks.ejs'. The 'confetti.ejs' tab has its content displayed below it. The 'thanks.ejs' tab is shown in a preview pane at the top right. The 'confetti.ejs' file contains a series of nested divs with classes like 'confetti-149', 'confetti-148', etc. A red box highlights the line of code '<% include('partials/confetti') %>' which includes the partial template. The 'thanks.ejs' file shows the main layout with a footer section.

thanks.ejs에서

```
<%- include('partials/confetti') %>
```

재사용 가능한 모든 코드를 별도의 파일에 복사하여 붙여넣습니다. 이렇게 하면 관리가 더 쉬워집니다. 그런 다음 하나의 파일만 모든 페이지 요소를 제어합니다. 코드를 변경하기 위해 모든 페이지를 수정할 필요는 없습니다. 해당 요소의 파일만 수정하면 됩니다.



```
// resources.md      <> thanks.ejs ✘ <> confetti.ejs <> courses.ejs M <> layout.ejs <>
과제 > solutions > midterm-test-soln > views > <> thanks.ejs > html > body > main. > ?
//           </div>
78          </div>
79
80  <% include('partials/confetti') %>
81  </main>
82
83  <!-- FOOTER -->
84  <footer class="text-center fixed-bottom bg-dark text-bg-dark pt-3">
85    <p class="ut-red">
86      <span class="text-white">&copy; 2023</span> [[ [ 학생의 이름과 학번 ] ]]
87    </p>
88  </footer>
89
90  <!-- Bootstrap의 JS -->
91  <script src="../../public/js/bootstrap.bundle.min.js"></script>
92  </body>
93  </html>
94
```

header.ejs 와 navigation.ejs

모든 페이지에 헤더와 탐색이 필요하지 않습니다. 대신 하나의 파일에만 포함하고 다른 모든 파일에는 해당 파일을 포함합니다.

EXPLORER ... resources.md thanks.ejs confetti.ejs courses.ejs M layout.ejs X header.ejs

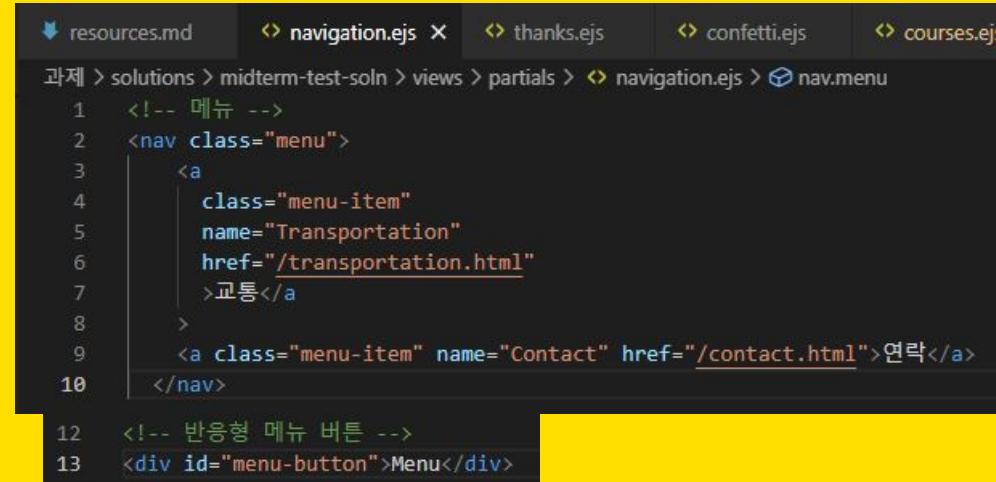
2023.1-UT-NODEJS

- > 3-bootstrap-soln
- > 3-express-webserver-soln
- > 4-express-capstone-soln
- > 4-mongodb-capstone-soln
- > 4-mongodb-soln
- > 5-crud-soln
- midterm-test-soln
 - > node_modules
 - > pdfs
 - > public
 - > views
 - > partials
 - ↳ confetti.ejs
 - ↳ footer.ejs
 - ↳ header.ejs
 - ↳ navigation.ejs
 - ↳ contact.ejs
 - ↳ error.ejs
 - ↳ index.ejs
 - ↳ layout.ejs
 - ↳ thanks.ejs
 - ↳ transportation.ejs
 - ↳ .gitignore
 - ↳ bonus.md
 - ↳ checklist.test.md
 - ↳ package-lock.json
 - ↳ package.json
 - ↳ readme.en.md
 - ↳ README.md
 - ↳ study-guide.md
 - > solutions-public
 - > 시험
 - ↳ EN-get-programming-with-node.js-j...
 - ↳ KO-get programming with node.js-K...
 - ↳ KO-get programming with node.js-K...
 - ↳ Node.js 교과서.pdf
 - ↳ README.md
 - ↳ 컴퓨터공학전공_웹프로그래밍응용_...
 - ↳ 컴퓨터공학전공_웹프로그래밍응용_...

25 <body id="index">
26 <!-- HEADER -->
27 <header class="header">
28 <div class="container">
29 <div class="container-left">
30 <!-- 로고 -->
31 <figure class="logo">
32
33 
34 <h1 class="site-title">
35 | UT-NodeJS 중간고사
36 | </h1>
37 |
38 </figure>
39 <!-- 메뉴 -->
40 <nav class="menu">
41 교통
42 연락
43 </nav>
44 <!-- 반응형 메뉴 버튼 -->
45 <div id="menu-button">Menu</div>
46 </div>
47 </div>
48 </header>
49 <!-- MAIN 페이지 내용 -->
50 <main id="index" class="text-center cover-container">
51 <div class="cover-content h-100 w-100 text-white">
52 <div class="cover-box-2">
53 <h1 class="page-title">
54 | UT-NodeJS 중간고사
55 | </h1>
56 <p class="page-description lead">
57 | I changed this simple site to an Express site.
58 | </p>
59 </div>
60 </div>
61 </main>

navigation.ejs

모든 페이지에 헤더와 탐색이 필요하지 않습니다. 대신 하나의 파일에만 포함하고 다른 모든 파일에는 해당 파일을 포함합니다.



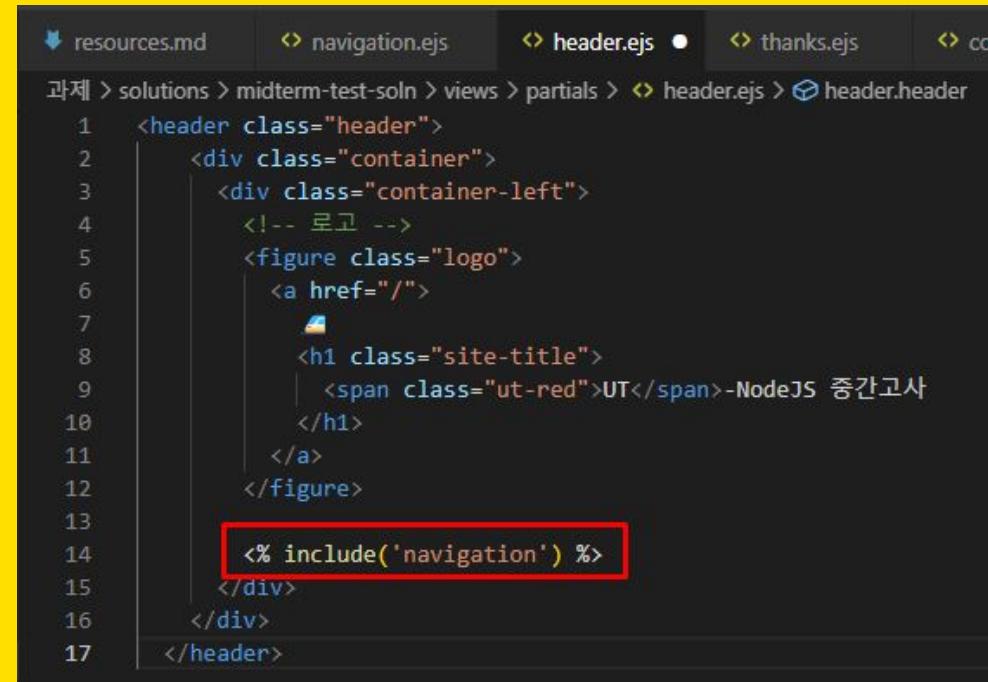
The screenshot shows a code editor window with several tabs at the top: resources.md, navigation.ejs (which is the active tab), thanks.ejs, confetti.ejs, and courses.ejs. The file path is displayed as 과제 > solutions > midterm-test-soln > views > partials > navigation.ejs > nav.menu. The code itself is as follows:

```
1  <!-- 메뉴 -->
2  <nav class="menu">
3      <a
4          class="menu-item"
5          name="Transportation"
6          href="/transportation.html"
7          >교통</a>
8      >
9      <a class="menu-item" name="Contact" href="/contact.html">연락</a>
10     </nav>
11
12    <!-- 반응형 메뉴 버튼 -->
13    <div id="menu-button">Menu</div>
```

header.ejs

```
<%- include('navigation') %>
```

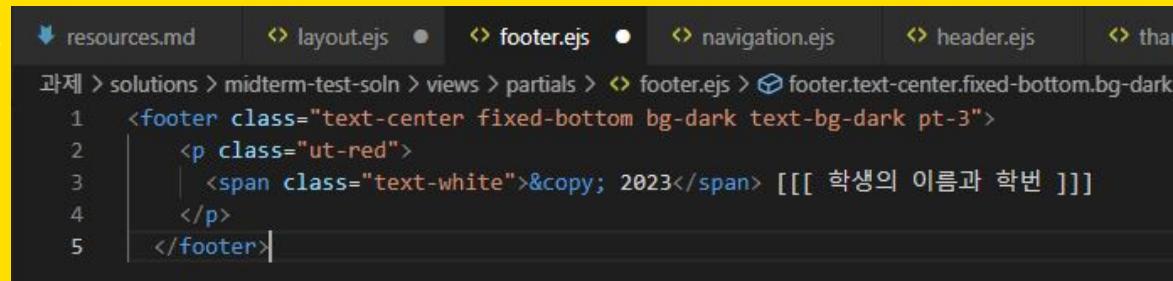
모든 페이지에 헤더와 탐색이 필요하지 않습니다. 대신 하나의 파일에만 포함하고 다른 모든 파일에는 해당 파일을 포함합니다.



```
1 <header class="header">
2   <div class="container">
3     <div class="container-left">
4       <!-- 로고 -->
5       <figure class="logo">
6         <a href="/">
7           &#x1f3a9;
8           <h1 class="site-title">
9             <span class="ut-red">UT</span>-NodeJS 중간고사
10            </h1>
11          </a>
12        </figure>
13
14        <% include('navigation') %>
15      </div>
16    </div>
17  </header>
```

footer.ejs

모든 페이지에 헤더와 탐색이 필요하지 않습니다. 대신 하나의 파일에만 포함하고 다른 모든 파일에는 해당 파일을 포함합니다.



The screenshot shows a code editor window with a yellow background. The title bar includes tabs for 'resources.md', 'layout.ejs', 'footer.ejs' (which is the active tab), 'navigation.ejs', 'header.ejs', and 'tha'. Below the title bar, the file path is displayed: '과제 > solutions > midterm-test-soln > views > partials > footer.ejs'. The code itself is as follows:

```
1  <footer class="text-center fixed-bottom bg-dark text-bg-dark pt-3">
2    <p class="ut-red">
3      <span class="text-white">&copy; 2023</span> [[ 학생의 이름과 학번 ]]
4    </p>
5  </footer>
```

layout.ejs

```
<%- include('partials/header') %>
<%- include('partials/footer') %>
```

이제 **<main>** 태그 사이의 모든 것이 페이지 콘텐츠입니다. 이것은 개별 페이지에 남아 있어야 하는 유일한 요소입니다. **<%- body %>** 기능을 사용하여 각 페이지의 내용을 가져올 수 있습니다.

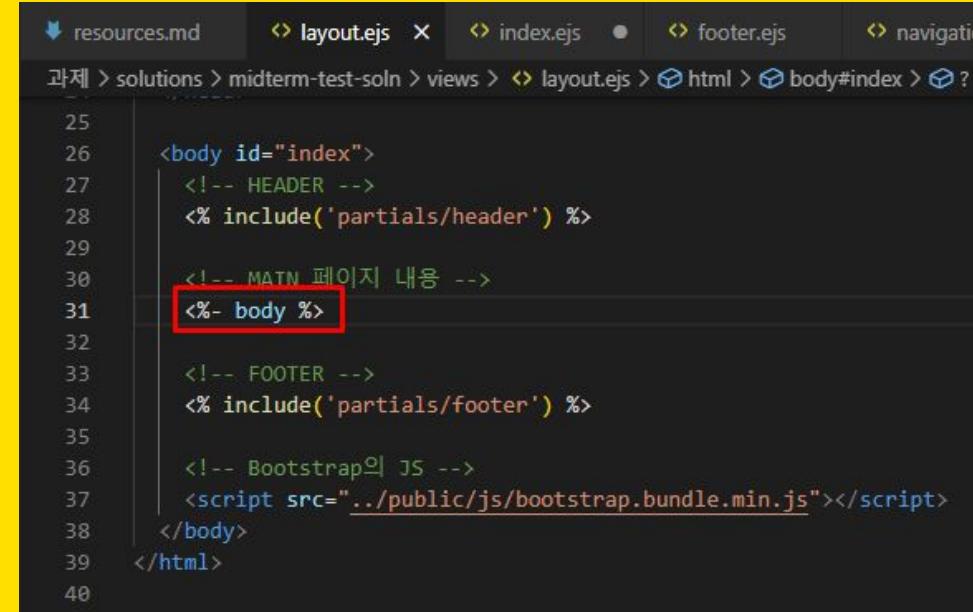
```
resources.md      layout.ejs X    footer.ejs    navigation.ejs    header.ejs
과제 > solutions > midterm-test-soln > views > layout.ejs > html > body#index > ? > ? ?

25
26   <body id="index">
27     <!-- HEADER -->
28     <% include('partials/header') %>
29
30     <!-- MAIN 페이지 내용 -->
31     <main id="index" class="text-center cover-container">
32       <div class="cover-content h-100 w-100 text-white">
33         <div class="cover-box-2">
34           <h1 class="page-title">
35             <span class="ut-red">UT</span>-NodeJS 중간고사
36           </h1>
37           <p class="page-description lead">
38             I changed this simple site to an Express site.
39           </p>
40         </div>
41       </div>
42     </main>
43
44     <!-- FOOTER -->
45     <% include('partials/footer') %>
46
47     <!-- Bootstrap의 JS -->
48     <script src="../public/js/bootstrap.bundle.min.js"></script>
49   </body>
50 </html>
51
```

layout.ejs

```
<%- include('partials/header') %>
<%- include('partials/footer') %>
```

이제 **<main>** 태그 사이의 모든 것이 페이지 콘텐츠입니다. 이것은 개별 페이지에 남아 있어야 하는 유일한 요소입니다. **<%- body %>** 기능을 사용하여 각 페이지의 내용을 가져올 수 있습니다.



```
resources.md layout.ejs index.ejs footer.ejs navigation
과제 > solutions > midterm-test-soln > views > layout.ejs > html > body#index > ??
25 <body id="index">
26   <!-- HEADER -->
27   <% include('partials/header') %>
28
29   <!-- MAIN 페이지 내용 -->
30   <%- body %> <%- body %>
31
32   <!-- FOOTER -->
33   <% include('partials/footer') %>
34
35   <!-- Bootstrap의 JS -->
36   <script src="../../public/js/bootstrap.bundle.min.js"></script>
37
38 </body>
39 </html>
40
```

index.ejs

```
resources.md      JS main.js U      {} package.json U      ⓘ readme.md      JS error
과제 > solutions > midterm-test-soln > views > index.ejs > main#index.text-center.cover-
  1  <main id="index" class="text-center cover-container">
  2    <div class="cover-content h-100 w-100 text-white">
  3      <div class="cover-box-2">
  4        <h1 class="page-title">
  5          <span class="ut-red">UT</span>-NodeJS 중간고사
  6        </h1>
  7        <p class="page-description lead">
  8          I changed this simple site to an Express site.
  9        </p>
 10      </div>
 11    </div>
 12  </main>
```

contact.ejs

```
resources.md      layout.ejs    index.ejs   contact.ejs ● footer.ejs   navigation.ejs  header.ejs
과제 > solutions > midterm-test-soln > views > contact.ejs > ...
1   <!-- MAIN 페이지 내용 -->
2   <main class="">
3       <div class="container col-xxl-10 col-xxl-8 px-4 py-5">
4           <div class="row align-items-center g-lg-5 py-3">
5               <div class="col-lg-7 text-center text-lg-start mb-3">
6                   <figure>
7                       
10                  </figure>
11                  <h1 class="display-4">Contact Us</h1>
12                  <p class="lead col-lg-10 fs-4">Get in touch to find out more.</p>
13              </div>
14
15          <div class="col-md-10 mx-auto col-lg-5 mb-5">
16              > <form...>
17                  </form>
18          </div>
19      </div>
20  </div>
21  </main>
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

thanks.ejs

```
resources.md      layout.ejs      thanks.ejs • index.ejs      contact.ejs      error.ejs      footer.ejs
과제 > solutions > midterm-test-soln > views > thanks.ejs > ...
1   <!-- MAIN 페이지 내용 -->
2   <main class="">
3       <div class="container col-xxl-10 col-xxl-8 px-4 py-5">
4           <div class="row align-items-center g-lg-5 py-5">
5               <div class="col-lg-7">
6                   <figure>
7                       
10                  </figure>
11              </div>
12
13          <div class="col-md-10 mx-auto col-lg-5 text-center text-lg-start">
14              <h1 class="display-4">Thanks!</h1>
15              <p class="lead col-lg-10 fs-4">We'll keep you posted!</p>
16          </div>
17      </div>
18  </div>
19
20  <% include('partials/confetti') %>
21  </main>
22
```

transportation.ejs

```
resources.md    layout.ejs    thanks.ejs    transportation.ejs ●    index.ejs    contact.ejs
과제 > solutions > midterm-test-soln > views > transportation.ejs > main.ejs > div#carouselExampleCaptions.carousel.slide >
  1 |   <!-- MAIN 페이지 내용 -->
  2 |   <main class="">
  3 |     <div
  4 |       id="carouselExampleCaptions"
  5 |       class="carousel slide"
  6 |       data-bs-ride="carousel"
  7 |     >
  8 |       <div class="carousel-indicators">
  9 |         <button ...
 10 |           ...
 11 |         ></button>
 12 |         <button ...
 13 |           ...
 14 |         ></button>
 15 |         <button ...
 16 |           ...
 17 |         ></button>
 18 |       </div>
 19 |       <div class="carousel-inner">
 20 |         <!-- 1. Carousel Item ONE -->
 21 |         <div class="carousel-item active">...
 22 |           ...
 23 |         </div>
 24 |
 25 |         <!-- 2. Carousel Item TWO -->
 26 |         <div class="carousel-item">...
 27 |           ...
 28 |         </div>
 29 |
 30 |         <!-- 3. Carousel Item THREE -->
 31 |         <div class="carousel-item">...
 32 |           ...
 33 |         </div>
 34 |
 35 |         <button ...
 36 |           ...
 37 |         ></button>
 38 |         <button ...
 39 |           ...
 40 |         ></button>
 41 |         <button ...
 42 |           ...
 43 |         ></button>
 44 |       </div>
 45 |
 46 |       <div class="text-center pt-5">
 47 |         <h1>Romantic Travel by Train</h1>
 48 |         <p class="lead">
 49 |           Discover why some of the most beautiful and memorable travel
 50 |           experiences happen on the rail.
 51 |         </p>
 52 |       </div>
 53 |
 54 |     </main>
```

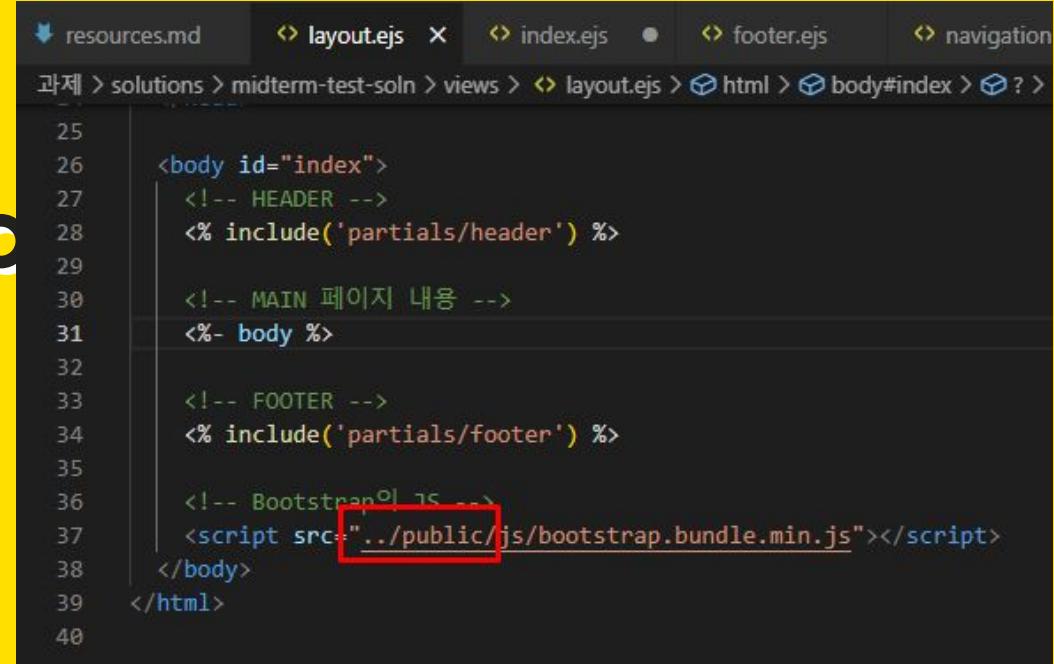
error.ejs

error.ejs 페이지에서 정적 이미지를 사용하고 있습니다. 그러나 명시적으로 URL에 "../public"을 포함할 필요는 없습니다. **error.ejs**와 **layout.ejs**에서 삭제할 수 있습니다.

```
resources.md      layout.ejs      index.ejs      contact.ejs      error.ejs      footer.ejs  
과제 > solutions > midterm-test-soln > views > error.ejs > ...  
1 | <!-- MAIN 페이지 내용 -->  
2 | <main class="">  
3 |   <div class="container px-4 py-5 text-center mb-5">  
4 |     <!-- Inspiration: https://codepen.io/abol/pen/arPeZz -->  
5 |     <h1 class="display-4">404</h1>  
6 |     <figure class="img-404">  
7 |       <img src='../public/img/dribbble_404.gif' />  
8 |     </figure>  
9 |     <h2>Looks like you're lost.</h2>  
10 |    <p class="pt-3 pb-3 w-75 mx-auto">  
11 |      The page you are looking for was not found. But you can click the  
12 |      button below to go back to the homepage.  
13 |    </p>  
14 |    <a href="/" class="btn btn-lg ut-red">Back to Home</a>  
15 |  </div>  
16 |</main>  
17 |
```

layout.ejs

error.ejs 페이지에서 정적 이미지를 사용하고 있습니다. 그러나 명시적으로 URL에 "../public"을 포함할 필요는 없습니다. **error.ejs**와 **layout.ejs**에서 삭제할 수 있습니다.



The screenshot shows a code editor window with the following file structure:

```
resources.md      layout.ejs ✘ index.ejs ● footer.ejs navigation
```

The current file is `layout.ejs`. The code content is as follows:

```
25 <body id="index">
26   <!-- HEADER -->
27   <% include('partials/header') %>
28
29   <!-- MAIN 페이지 내용 -->
30   <%- body %>
31
32   <!-- FOOTER -->
33   <% include('partials/footer') %>
34
35   <!-- Bootstrap CSS -->
36   <script src="../public/js/bootstrap.bundle.min.js"></script>
37
38 </body>
39 </html>
40
```

A red rectangle highlights the line `<script src="../public/js/bootstrap.bundle.min.js"></script>`, indicating it is the target for modification.

Update layout.ejs

layout.ejs의 모든 URL에서 "../public"을 삭제한 다음 작성자(귀하의) 정보도 업데이트합니다.

```
resources.md layout.ejs index.ejs footer.ejs navigation.ejs
과제 > solutions > midterm-test-soln > views > layout.ejs > html > head > link
1  <!-- UT-NodeJS 중간고사 -->
2  <!DOCTYPE html>
3  <html lang="en">
4      <head>
5          <meta charset="UTF-8" />
6          <meta name="viewport" content="width=device-width, initial-scale=1" />
7          <title>UT-NodeJS 중간고사 | 학생 이름</title>
8
9          <!-- 학생 이름과 학번 -->
10         <meta name="author" content="????" />
11         <meta name="description" content="????" />
12
13         <!-- CSS 파일에 대한 link태그를 수정하십시오 -->
14         <link href="/css/bootstrap.min.css" rel="stylesheet" />
15         <link rel="stylesheet" href="/css/style.css"/>
16         <link
17             rel="stylesheet"
18             media="(max-width: 768px)"
19             href="/css/style.small.css"
20         />
21
22         <!-- JS 파일에 대한 script태그를 수정하십시오 -->
23         <script defer src="/js/functions.js"></script>
24     </head>
25
26     <body id="index">
27         <!-- HEADER -->
28         <% include('partials/header') %>
29
30         <!-- MAIN 페이지 내용 -->
31         <%- body %>
32
33         <!-- FOOTER -->
34         <% include('partials/footer') %>
35
36         <!-- Bootstrap의 JS -->
37         <script src="/js/bootstrap.bundle.min.js"></script>
38     </body>
39 </html>
```

Update layout.ejs

layout.ejs의 모든 URL에서 "../public"을 삭제한 다음 작성자(귀하의) 정보도 업데이트합니다.

```
resources.md layout.ejs index.ejs footer.ejs navigation.ejs
과제 > solutions > midterm-test-soln > views > layout.ejs > html > head
1  <!-- UT-NodeJS 중간고사 -->
2  <!DOCTYPE html>
3  <html lang="en">
4    <head>
5      <meta charset="UTF-8" />
6      <meta name="viewport" content="width=device-width, initial-scale=1" />
7      <title>UT-NodeJS 중간고사 | Aaron Snowberger</title>
8
9      <!-- 학생 이름과 학번 -->
10     <meta name="author" content="Aaron Snowberger" />
11     <meta name="description" content="Midterm Test" />
12
13     <!-- CSS 파일에 대한 link태그를 수정하십시오 -->
14     <link href="/css/bootstrap.min.css" rel="stylesheet" />
15     <link rel="stylesheet" href="/css/style.css" />
16     <link
17       rel="stylesheet"
18       media="(max-width: 768px)"
19       href="/css/style.small.css"
20     />
21
22     <!-- JS 파일에 대한 script태그를 수정하십시오 -->
23     <script defer src="/js/functions.js"></script>
24   </head>
25
26   <body id="index">
27     <!-- HEADER -->
28     <% include('partials/header') %>
29
30     <!-- MAIN 페이지 내용 -->
31     <%- body %>
32
33     <!-- FOOTER -->
34     <% include('partials/footer') %>
35
36     <!-- Bootstrap의 JS -->
37     <script src="/js/bootstrap.bundle.min.js"></script>
38   </body>
39 </html>
```

03

웹 앱 만들기

main.js
homeController.js
errorController.js

main.js

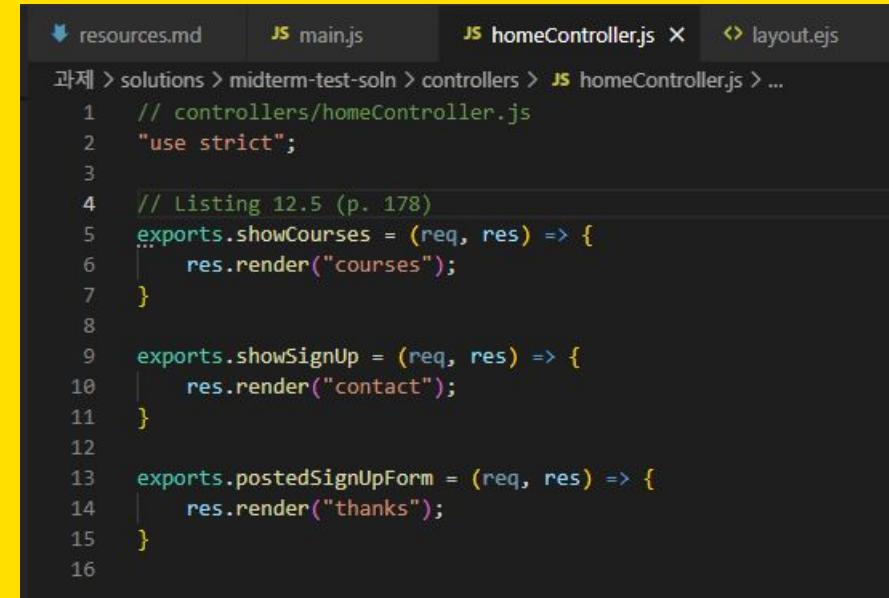
웹 서버를 만들고 실행하는 대부분의 코드는 Listing 12.3, 12.4, 12.5, 그리고 12.7에서 찾을 수 있습니다.

```
resources.md JS main.js X layout.ejs X thanks.ejs X transportation.ejs
과제 > solutions > midterm-test-soln > JS main.js > ...
1 // main.js
2 "use strict";
3
4 // Listing 12.3 (p. 176-177)
5 const port = 3000,
6     layouts = require('express-ejs-layouts'), // Listing 12.7 (p. 179)
7     express = require("express"),
8     app = express();
9
10 app.set("port", process.env.PORT || port);
11
12 // Listing 12.7 (p. 179)
13 app.set("view engine", "ejs");
14 app.use(layouts);
15
16 app.use(express.static("public")); // 제목 12.5 (p. 181)
17
18 // Listing 12.4 (p. 177)
19 app.use(express.urlencoded({ extended: false }));
20 app.use(express.json());
21
22 // ...
23
24 app.listen(app.get("port"), () => {
25   console.log(`Server running at: http://localhost:\${app.get\("port"\)}`);
26 })
```

homeController.js

homeController에 대한 코드는 Listing 12.5에서 찾을 수 있습니다.

그러나 포함된 "인덱스" 경로가 없으므로 하나를 생성해야 합니다.



The screenshot shows a code editor window with several tabs at the top: resources.md, main.js, homeController.js (which is the active tab), and layout.ejs. The file path in the title bar is: 과제 > solutions > midterm-test-soln > controllers > homeController.js. The code in the editor is as follows:

```
// controllers/homeController.js
"use strict";

// Listing 12.5 (p. 178)
exports.showCourses = (req, res) => {
    res.render("courses");
}

exports.showSignUp = (req, res) => {
    res.render("contact");
}

exports.postedSignUpForm = (req, res) => {
    res.render("thanks");
}
```

homeController.js

homeController에 대한 코드는 Listing 12.5에서 찾을 수 있습니다.

그러나 포함된 "인덱스" 경로가 없으므로 하나를 생성해야 합니다.

또한 "courses"를 "transportation"으로 바꿔야 합니다.

```
// controllers/homeController.js
"use strict";

exports.showHome = (req, res) => {
    res.render("index");
};

// Listing 12.5 (p. 178)
exports.showTransportation = (req, res) => {
    res.render("transportation");
};

exports.showSignUp = (req, res) => {
    res.render("contact");
};

exports.postedSignUpForm = (req, res) => {
    res.render("thanks");
};
```

main.js

main.js에 homeController 코드 포함.

```
resources.md JS main.js X JS homeController.js < layout.ejs < thanks.ejs < tran  
과제 > solutions > midterm-test-soln > JS main.js > ...  
1 // main.js  
2 "use strict";  
3  
4 // Listing 12.3 (p. 176-177)  
5 const port = 3000,  
6     layouts = require('express-ejs-layouts'), // Listing 12.7 (p. 179)  
7     express = require("express"),  
8     app = express();  
9  
10 app.set("port", process.env.PORT || port);  
11  
12 // Listing 12.7 (p. 179)  
13 app.set("view engine", "ejs");  
14 app.use(layouts);  
15  
16 app.use(express.static("public")); // 제목 12.5 (p. 181)  
17  
18 // Listing 12.4 (p. 177)  
19 app.use(express.urlencoded({ extended: false }));  
20 app.use(express.json());  
21  
22 // homeController.js  
23 const homeController = require("./controllers/homeController");  
24  
25 // Listing 12.6 (p. 178)  
26 app.get("/", homeController.showHome);  
27 app.get("/transportation", homeController.showTransportation);  
28 app.get("/contact", homeController.showSignUp);  
29 app.post("/contact", homeController.postedSignUpForm);  
30  
31 app.listen(app.get("port"), () => {  
32     console.log(`Server running at: http://localhost:\${app.get\("port"\)}`);  
33 });
```

errorController.js

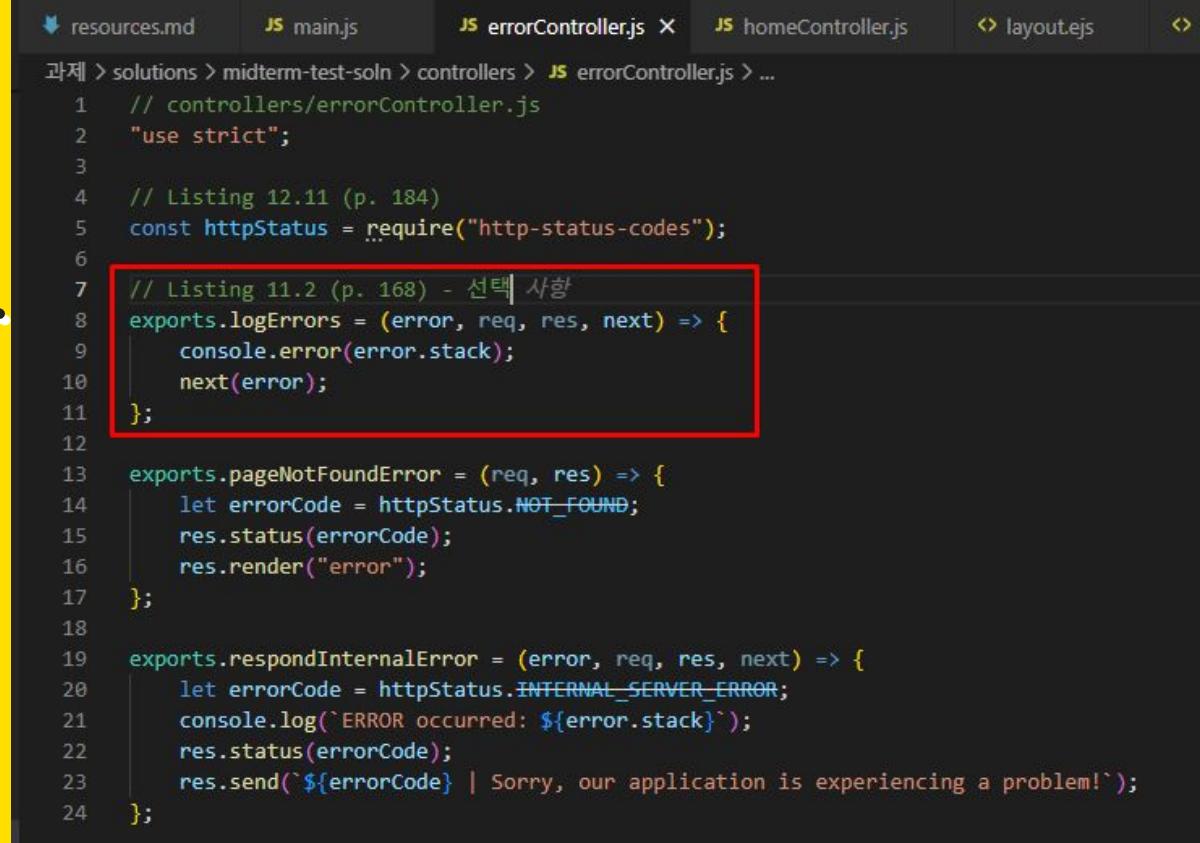
errorController에 대한 코드는 Listing 12.11에서 찾을 수 있습니다.

```
resources.md main.js errorController.js X homeController.js layout.ejs  
과제 > solutions > midterm-test-soln > controllers > errorController.js > ...  
1 // controllers/errorController.js  
2 "use strict";  
3  
4 // Listing 12.11 (p. 184)  
5 const httpStatus = require("http-status-codes");  
6  
7 exports.pageNotFoundError = (req, res) => {  
8     let errorCode = httpStatus.NOT_FOUND;  
9     res.status(errorCode);  
10    res.render("error");  
11};  
12  
13 exports.respondInternalError = (error, req, res, next) => {  
14     let errorCode = httpStatus.INTERNAL_SERVER_ERROR;  
15     console.log(`ERROR occurred: ${error.stack}`);  
16     res.status(errorCode);  
17     res.send(`${errorCode} | Sorry, our application is experiencing a problem!`);  
18};
```

errorController.js

errorController에 대한 코드는 Listing 12.11에서 찾을 수 있습니다.

오류 로깅 기능을 포함하려는 경우
Listing 11.2에서 찾을 수 있습니다.



```
// controllers/errorController.js
"use strict";

// Listing 12.11 (p. 184)
const httpStatus = require("http-status-codes");

// Listing 11.2 (p. 168) - 선택 사항
exports.logErrors = (error, req, res, next) => {
  console.error(error.stack);
  next(error);
};

exports.pageNotFoundError = (req, res) => {
  let errorCode = httpStatus.NOT_FOUND;
  res.status(errorCode);
  res.render("error");
};

exports.respondInternalError = (error, req, res, next) => {
  let errorCode = httpStatus.INTERNAL_SERVER_ERROR;
  console.log(`ERROR occurred: ${error.stack}`);
  res.status(errorCode);
  res.send(`${errorCode} | Sorry, our application is experiencing a problem!`);
};
```

main.js

main.js와 errorController 코드 포함.

```
resources.md JS main.js X JS errorController.js JS homeController.js < layout.ejs < tha
과제 > solutions > midterm-test-soln > JS main.js > ...
1 // main.js
2 "use strict";
3
4 // Listing 12.3 (p. 176-177)
5 const port = 3000,
6     layouts = require('express-ejs-layouts'), // Listing 12.7 (p. 179)
7     express = require("express"),
8     app = express();
9
10 app.set("port", process.env.PORT || port);
11
12 // Listing 12.7 (p. 179)
13 app.set("view engine", "ejs");
14 app.use(layouts);
15
16 app.use(express.static("public")); // 제목 12.5 (p. 181)
17
18 // Listing 12.4 (p. 177)
19 app.use(express.urlencoded({ extended: false }));
20 app.use(express.json());
21
22 // homeController.js
23 const homeController = require("./controllers/homeController");
24
25 // Listing 12.6 (p. 178)
26 app.get("/", homeController.showHome);
27 app.get("/transportation", homeController.showTransportation);
28 app.get("/contact", homeController.showSignUp);
29 app.post("/contact", homeController.postedSignUpForm);
30
31 // errorController.js
32 const errorController = require("./controllers/errorController");
33
34 // Listing 12.12 (p. 184)
35 app.use(errorController.logErrors);
36 app.use(errorController.respondNoResourceFound);
37 app.use(errorController.respondInternalError);
38
39 app.listen(app.get("port"), () => {
40   console.log(`Server running at: http://localhost:${app.get("port")}`);
41 });


```

04

실행!

코드 완성!

이제 실행하고 오류를
수정해 보겠습니다.

npm start 하면

이런, "start" 스크립트를 만드는 것을
잊은 것 같습니다.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    COMMENTS

found 0 vulnerabilities
(base) PS C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln> npm start
npm ERR! Missing script: "start"
npm ERR!
npm ERR! Did you mean one of these?
npm ERR!   npm star # Mark your favorite packages
npm ERR!   npm stars # View packages marked as favorites
npm ERR!
npm ERR! To see a list of scripts, run:
npm ERR!   npm run

npm ERR! A complete log of this run can be found in:
npm ERR!   C:\Users\ADMIN\AppData\Local\npm-cache\_logs\2023-04-27T02_17_02_183Z-debug-0.log
(base) PS C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln>
```

npm start 하면

이런, "start" 스크립트를 만드는 것을
잊은 것 같습니다.

```
resources.md JS main.js {} package.json X JS errorController.js JS
과제 > solutions > midterm-test-soln > {} package.json > {} scripts > start
1  {
2    "name": "midterm-test",
3    "version": "1.0.0",
4    "description": "Midterm Test for Aaron",
5    "main": "main.js",
6    "scripts": [
7      "test": "echo \\\"Error: no test specified\\\" && exit 1",
8      "start": "node main.js"
9    ],
10   "repository": {
11     "type": "git",
12     "url": "git+https://github.com/ut-nodejs/midterm-test.git"
13   },
14   "author": "Aaron Snowberger",
15   "license": "ISC",
16   "bugs": {
17     "url": "https://github.com/ut-nodejs/midterm-test/issues"
18   },
19   "homepage": "https://github.com/ut-nodejs/midterm-test#readme",
20   "dependencies": {
21     "ejs": "^3.1.9",
22     "express": "^4.18.2",
23     "express-ejs-layouts": "^2.5.1",
24     "http-status-codes": "^2.2.0"
25   }
26 }
27 }
```

에러 고치기

이제 실행되지만 오류가 있습니다.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

throw new TypeError('app.use() requires a middleware function')
^

TypeError: app.use() requires a middleware function
at Function.use (C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln\node_modules\express\lib\application.js:217:11)
at Object.<anonymous> (C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln\main.js:36:5)
at Module._compile (node:internal/modules/cjs/loader:1218:14)
at Module._extensions..js (node:internal/modules/cjs/loader:1272:10)
at Module.load (node:internal/modules/cjs/loader:1081:32)
at Module._load (node:internal/modules/cjs/loader:922:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:82:12)
at node:internal/main/run_main_module:23:47

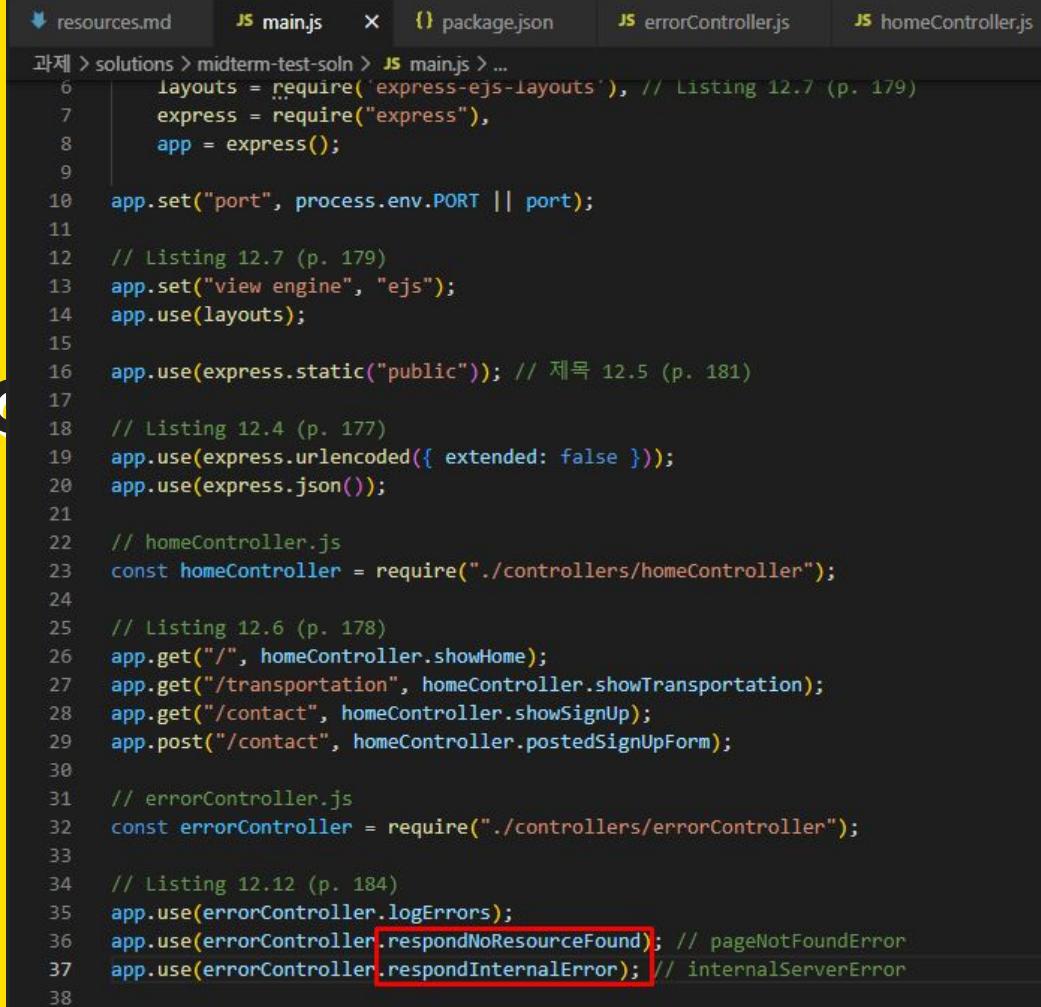
Node.js v19.3.0
(base) PS C:\Users\ADMIN\Desktop\UT\Classes\2023.1-UT-NodeJS\과제\solutions\midterm-test-soln>
```

에러 위치
main.js, line 36, char 5

main.js 예러 확인

이제 실행되지만 오류가 있습니다.
main.js, line 36, char 5

함수 이름의 철자가 잘못되었습니다.
철자를 항상 다시 확인하십시오.



The screenshot shows a code editor window with several tabs at the top: resources.md, main.js, package.json, errorController.js, and homeController.js. The main.js tab is active. The code editor displays the following JavaScript code:

```
↓ resources.md      JS main.js × {} package.json    JS errorController.js    JS homeController.js
과제 > solutions > midterm-test-soln > JS main.js > ...
6   layouts = require('express-ejs-layouts'), // Listing 12.7 (p. 179)
7   express = require("express"),
8   app = express();
9
10 app.set("port", process.env.PORT || port);
11
12 // Listing 12.7 (p. 179)
13 app.set("view engine", "ejs");
14 app.use(layouts);
15
16 app.use(express.static("public")); // 제목 12.5 (p. 181)
17
18 // Listing 12.4 (p. 177)
19 app.use(express.urlencoded({ extended: false }));
20 app.use(express.json());
21
22 // homeController.js
23 const homeController = require("./controllers/homeController");
24
25 // Listing 12.6 (p. 178)
26 app.get("/", homeController.showHome);
27 app.get("/transportation", homeController.showTransportation);
28 app.get("/contact", homeController.showSignUp);
29 app.post("/contact", homeController.postedSignUpForm);
30
31 // errorController.js
32 const errorController = require("./controllers/errorController");
33
34 // Listing 12.12 (p. 184)
35 app.use(errorController.logErrors);
36 app.use(errorController.respondNoResourceFound); // pageNotFoundError
37 app.use(errorController.respondInternalError); // internalServerError
38
```

A red rectangular box highlights the line of code: `app.use(errorController.respondNoResourceFound); // pageNotFoundError`. The code editor has a dark theme with syntax highlighting in green, blue, and orange.

localhost:3000

이제 사이트가 실행되지만 머리글이나
바닥글이 없습니다.

이런, <%- include %>에는 - 가
필요합니다.



Contact Us

Get in touch to find out more.

Subscribe to Newsletter

Submit

We'll never share your email with
anyone else.

localhost:3000

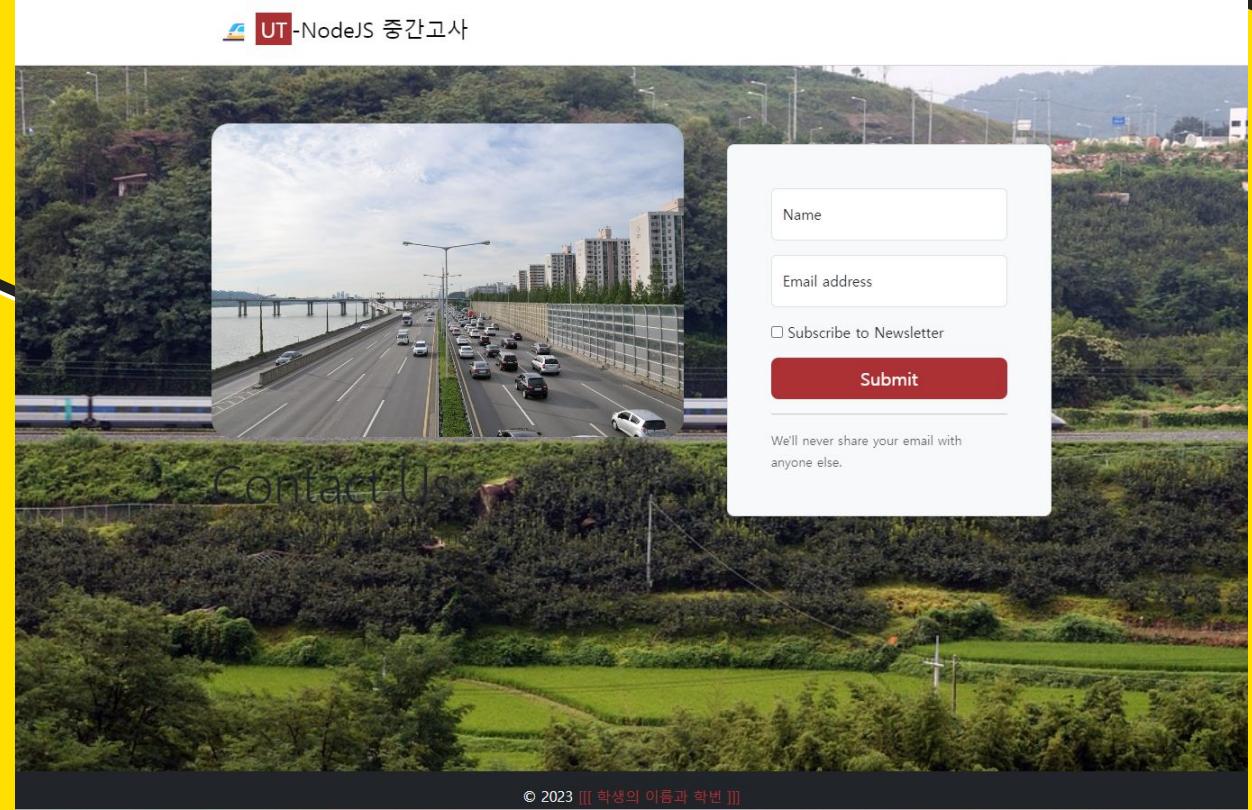
이제 사이트가 실행되지만 머리글이나 바닥글이 없습니다.

이런, <%- include %>에는 - 가 필요합니다.

```
resources.md      JS main.js      {} package.json      JS errorController.js
과제 > solutions > midterm-test-soln > views > layout.ejs > html > body > ? >
6   <meta name="viewport" content="width=device-width, initial-s
7     <title>UT-NodeJS 중간고사 | Aaron Snowberger</title>
8
9   <!-- 학생 이름과 학번 -->
10  <meta name="author" content="Aaron Snowberger" />
11  <meta name="description" content="Midterm Test" />
12
13  <!-- CSS 파일에 대한 link태그를 수정하십시오 -->
14  <link href="/css/bootstrap.min.css" rel="stylesheet" />
15  <link rel="stylesheet" href="/css/style.css" />
16  <link
17    rel="stylesheet"
18    media="(max-width: 768px)"
19    href="/css/style.small.css"
20  />
21
22  <!-- JS 파일에 대한 script태그를 수정하십시오 -->
23  <script defer src="/js/functions.js"></script>
24  </head>
25
26  <body>
27    <!-- HEADER -->
28    <%- include('partials/header') %>
29
30    <!-- MAIN 페이지 내용 -->
31    <%- body %>
32
33    <!-- FOOTER -->
34    <%- include('partials/footer') %>
35
36    <!-- Bootstrap의 JS -->
37    <script src="/js/bootstrap.bundle.min.js"></script>
38  </body>
```

localhost:3000

또한 이 사이트에는 모든 페이지에 배경 이미지가 나타나도록 하는 layout.ejs의 <body> 태그에 추가 CSS id="index"가 있습니다. 삭제할 수 있습니다.



localhost:3000

또한 이 사이트에는 모든 페이지에 배경 이미지가 나타나도록 하는 layout.ejs의 <body> 태그에 추가 CSS id="index"가 있습니다. 삭제할 수 있습니다.

```
resources.md      JS main.js      {} package.json      JS errorController.js      JS homeController.js
과제 > solutions > midterm-test-soln > views > layout.ejs > html > body#index
6   <meta name="viewport" content="width=device-width, initial-scale=1" />
7   <title>UT-NodeJS 중간고사 | Aaron Snowberger</title>
8
9   <!-- 학생 이름과 학번 -->
10  <meta name="author" content="Aaron Snowberger" />
11  <meta name="description" content="Midterm Test" />
12
13  <!-- CSS 파일에 대한 link태그를 수정하십시오 -->
14  <link href="/css/bootstrap.min.css" rel="stylesheet" />
15  <link rel="stylesheet" href="/css/style.css" />
16  <link
17    rel="stylesheet"
18    media="(max-width: 768px)"
19    href="/css/style.small.css"
20  />
21
22  <!-- JS 파일에 대한 script태그를 수정하십시오 -->
23  <script defer src="/js/functions.js"></script>
24 </head>
25
26 <body id="index">
27  <!-- HEADER -->
28  <%- include('partials/header') %>
29
30  <!-- MAIN 페이지 내용 -->
31  <%- body %>
32
33  <!-- FOOTER -->
34  <%- include('partials/footer') %>
35
36  <!-- Bootstrap의 JS -->
37  <script src="/js/bootstrap.bundle.min.js"></script>
38 </body>
```

웹 앱 완성!

이제 프로젝트가 완전히 실행됩니다.

Listing 12.9 또는 12.10의 코드를 사용할
필요는 없었지만 사용했다면 괜찮습니다.

다음에 MongoDB에서 데이터로 작업할 것입니다.