

9월 22일 (수)

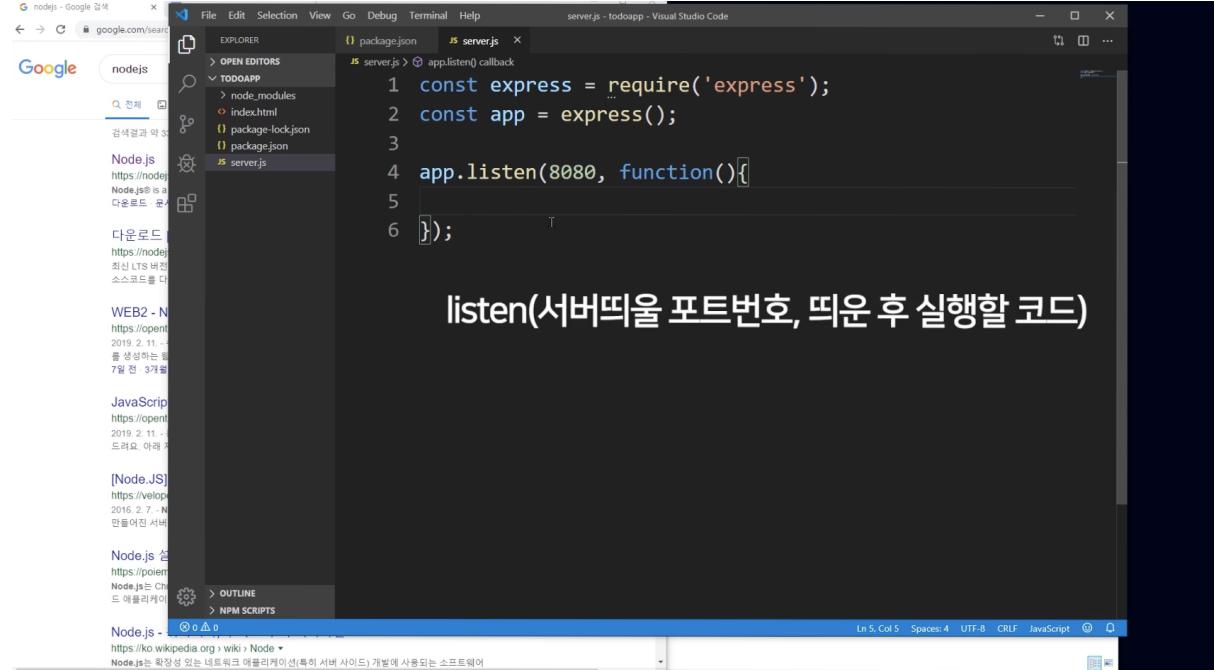
1. 코딩애플 - Node.js & MongoDB - Part 1.

6) 내 첫서버에서 GET 요청을 처리해보자

- Node.js Express 라이브러리에서 서버를 띄우기 위한 기본 설정

```
const express = require('express');
const app = express();

app.listen();
```



The screenshot shows a Windows desktop environment. In the foreground, the Visual Studio Code application is open, displaying a file named 'server.js' with the following code:

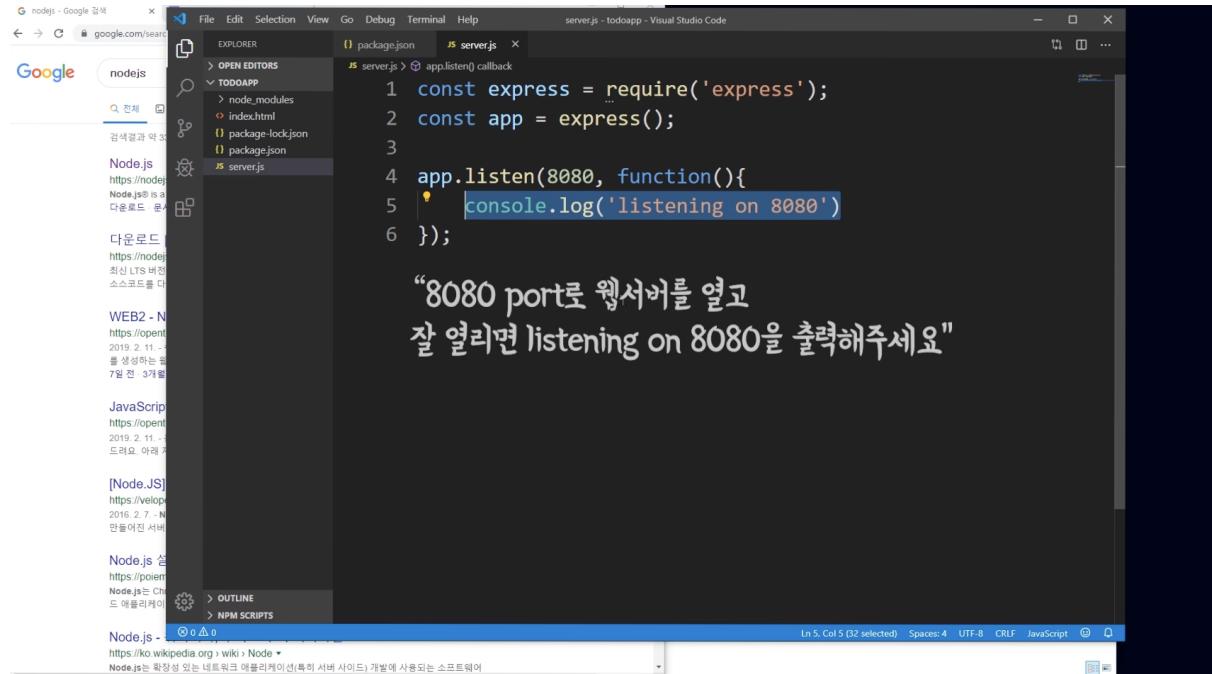
```
const express = require('express');
const app = express();

app.listen();
```

The code editor has syntax highlighting for JavaScript. In the background, a web browser window titled 'nodejs - Google 검색' is visible, showing search results for 'nodejs'. One result is from 'nodejs.org' with the URL 'https://nodejs.org'. Another result is from 'Wikipedia' with the URL 'https://ko.wikipedia.org/wiki/Node.js'.

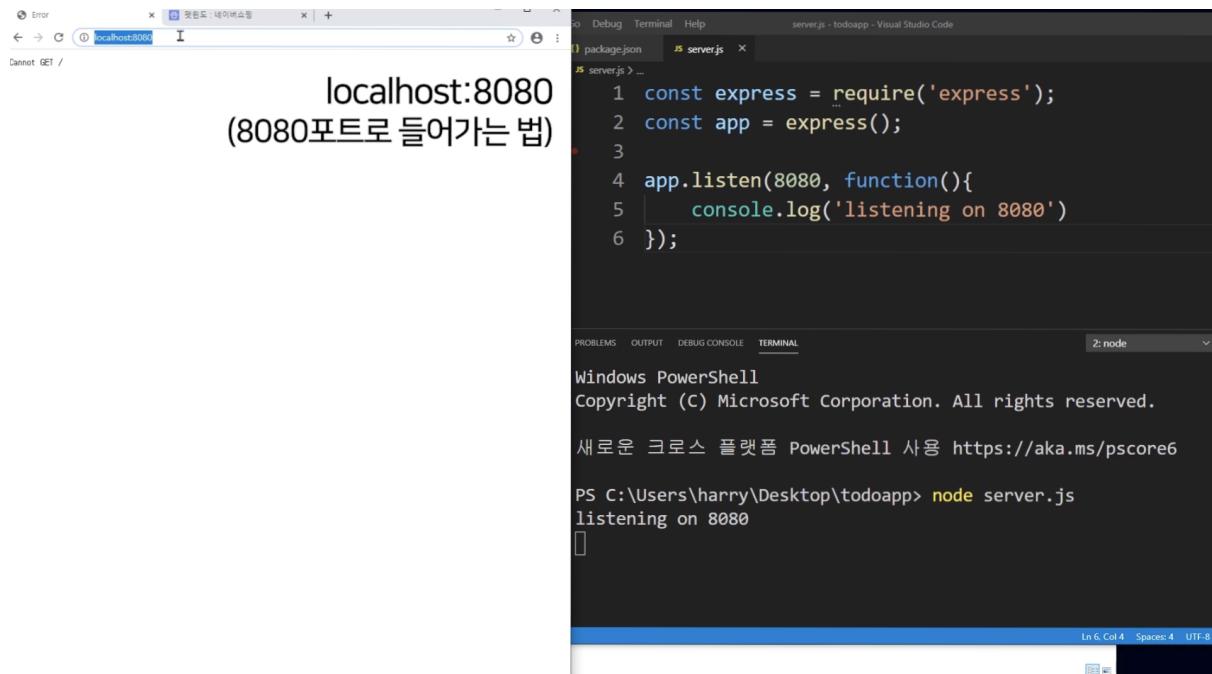
listen(서버띄울 포트번호, 띄운 후 실행할 코드)

포트란? 컴퓨터에는 외부 네트워크와 통신하기 위한 6만개 정도의 구멍(통로)이다. 위 코드는 6만개의 통로중 8080이라는 서버로 들어온 사람들에게 해당 함수를 실행해달라는 코드이다.



```
const express = require('express');
const app = express();

app.listen(8080, function(){
  console.log('listening on 8080')
});
```



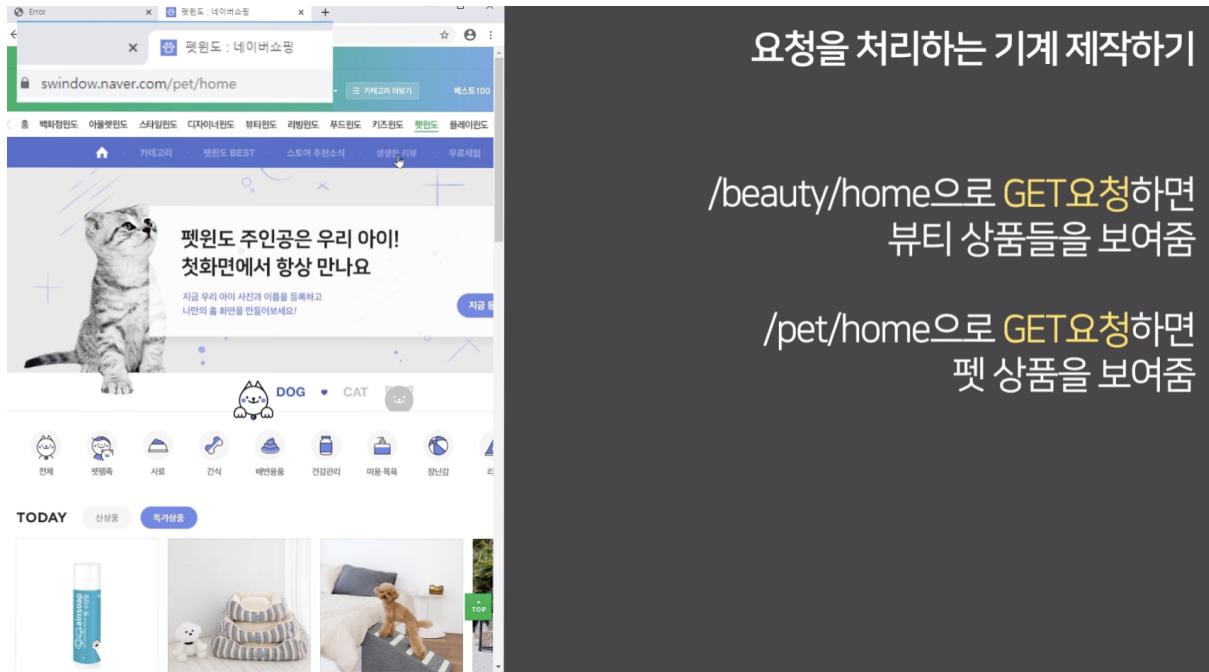
```
const express = require('express');
const app = express();

app.listen(8080, function(){
  console.log('listening on 8080')
});
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/powershell

PS C:\Users\harry\Desktop\todoapp> node server.js
listening on 8080
```



GET 요청을 처리하는
기계 제작하기

```

File Edit Selection View Go Debug Terminal Help
OPEN EDITORS JS server.js > ...
TODOAPP
node_modules
index.html
package-lock.json
package.json
server.js 7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\harry\Desktop\todoapp> node server.js
listening on 8080

```

The screenshot shows a Visual Studio Code interface with the following details:

- Terminal:** Displays "Windows PowerShell" and the command "PS C:\Users\harry\Desktop\todoapp> node server.js listening on 8080".
- Editor:** Shows the file "server.js" with code related to handling a GET request for "/pet".

```
6 });
7
8 누군가가 /pet 으로 방문을 하면..
9 pet관련된 안내문을 띄워주자
10
11 app.get('/pet', function(요청, 응답){
12     응답.send('반갑습니다');
13 });
14
15
16
```
- Output:** Shows "server.js - todoapp - Visual Studio Code" and "PROBLEMS: 0".
- Explorer:** Shows a project structure with files like package.json, index.html, package-lock.json, and server.js.

7) 서버에서 HTML 파일 전송해보기 & Nodemon으로 자동화

```

8 //누군가가 /pet 으로 방문을 하면..
9 //pet관련된 안내문을 띄워주자
10
11 app.get('/pet', function(요청, 응답){
12     응답.send('펫용품 쇼핑할 수 있는 페이지입니다.');
13 });
14
15
16

```

고객 (client) :
주소창에 URL을 입력해서
서버에 GET요청을 할 수 있음

서버 :
누군가 /pet으로 들어오면
XX를 보내주세요~ 라고 코드 짬

```

9 //pet관련된 안내문을 띄워주자
10
11 app.get('/pet', function(요청, 응답){
12     응답.send('펫용품 쇼핑할 수 있는 페이지입니다.');
13 });
14
15 app.get('/beauty', function(요청, 응답){
16     응답.send('뷰티용품 쇼핑할 수 있는 페이지입니다.');
17 });
18
19

```

.get() 여러개로 경로를 많이 생성 가능

새로운 크로스 플랫폼 PowerShell 사용 <https://aka.ms/powershell>

```

PS C:\Users\harry\Desktop\todoapp> node server.js
listening on 8080
PS C:\Users\harry\Desktop\todoapp> node server.js
listening on 8080

```

뷰티용품 쇼핑할 수 있는 페이지입니다.

```

10
11 app.get('/pet', function(요청, 응답){
12     응답.send('펫용품 쇼핑할 수 있는 페이지입니다.');
13 });
14
15 app.get('/beauty', function(요청, 응답){
16     응답.send('뷰티용품 쇼핑할 수 있는 페이지입니다.');
17 });
18
19
▼ 서버 재실행 귀찮음 (자동화시키자)

```

```

PS C:\Users\harry\Desktop\todoapp> node server.js
listening on 8080

```

뷰티용품 쇼핑할 수 있는 페이지입니다.

**nodemon으로
서버 재실행
자동화하기**

- npm으로 설치

```

10
11 app.get('/pet', function(요청, 응답){
12     응답.send('펫용품 쇼핑할 수 있는 페이지입니다.');
13 });
14
15 app.get('/beauty', function(요청, 응답){
16     응답.send('뷰티용품 쇼핑할 수 있는 페이지입니다.');
17 });
18
19

```

```

listening on 8080
PS C:\Users\harry\Desktop\todoapp> node server.js
listening on 8080
PS C:\Users\harry\Desktop\todoapp> node server.js
listening on 8080
PS C:\Users\harry\Desktop\todoapp> npm install -g nodemon
C:\Users\harry\AppData\Roaming\npm\nodemon -> C:\Users\harry\AppData\Roaming\npm\node_modules\nodemon\bin\nodemon.js
▼ 이렇게 설치하셈

```

Node.js 명령어에서 " -g " 의 뜻은 컴퓨터 모든 폴더에서 이용할 수 있도록 설치해주는 커맨드이다.

A screenshot of a browser window showing a simple HTML page with the title "/어쩌구로 접속시 HTML 파일을 보내보자" and a message "뷰티용품 쇼핑할 수 있는 페이지입니다.". Below it is a Visual Studio Code interface. The terminal tab shows the command `node server.js` being run, followed by the output: `mon] restarting due to changes... mon] starting 'node server.js' listening on 8080`.

```
app.get('/pet', function(요청, 응답){  
    응답.send('펫용품 쇼핑할 수 있는 페이지입니다.');//  
});  
  
app.get('/beauty', function(요청, 응답){  
    응답.send('뷰티용품 쇼핑할 수 있는 페이지입니다.');//  
});
```

A screenshot of a browser window showing the same simple HTML page. Below it is a Visual Studio Code interface. The terminal tab shows the command `node server.js` being run, followed by the output: `[nodemon] restarting due to changes... [nodemon] starting 'node server.js' listening on 8080`.

```
15 app.get('/beauty', function(요청, 응답){  
16     응답.send('뷰티용품 쇼핑할 수 있는 페이지입니다.');//  
17 });  
18  
19 app.get('/', function(요청, 응답){  
20     응답.send('뷰티용품 쇼핑할 수 있는 페이지입니다.');//  
21 });//  
22  
23
```

Visual Studio Code interface showing a Node.js application. The browser window displays the message "뷰티용품 쇼핑할 수 있는 페이지입니다." (Beauty products shopping page). The code editor shows server.js with the following content:

```
15 app.get('/beauty', function(요청, 응답){  
16     응답.send('뷰티용품 쇼핑할 수 있는 페이지입니다.');//  
17 });  
18  
19 app.get('/', function(요청, 응답){  
20     응답.sendFile(__dirname + '/index.html')  
21 });  
22  
23
```

The terminal shows the application starting on port 8080.

Visual Studio Code interface showing a Node.js application. The browser window displays the message "안녕하세요" (Hello) and "줄입니다." (This is a line). The code editor shows index.html with the following content:

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4     <meta charset="UTF-8">  
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">  
7     <title>Document</title>  
8 </head>  
9 <body>  
10    <h4>안녕하세요</h4>  
11    <p>줄입니다.</p>  
12 </body>  
13 </html>
```

The terminal shows the application starting on port 8080. A watermark "HTML로 꾸미면 멋진 사이트 완성" is visible on the right side of the browser window.