
HTML에서 웹앱까지

11주차_01

한 동 대 학 교
김군오 교수

학습 목표:

npm과 express 모듈 알아보기

- NPM로 모듈 설치
- Express.js 모듈 사용

NPM

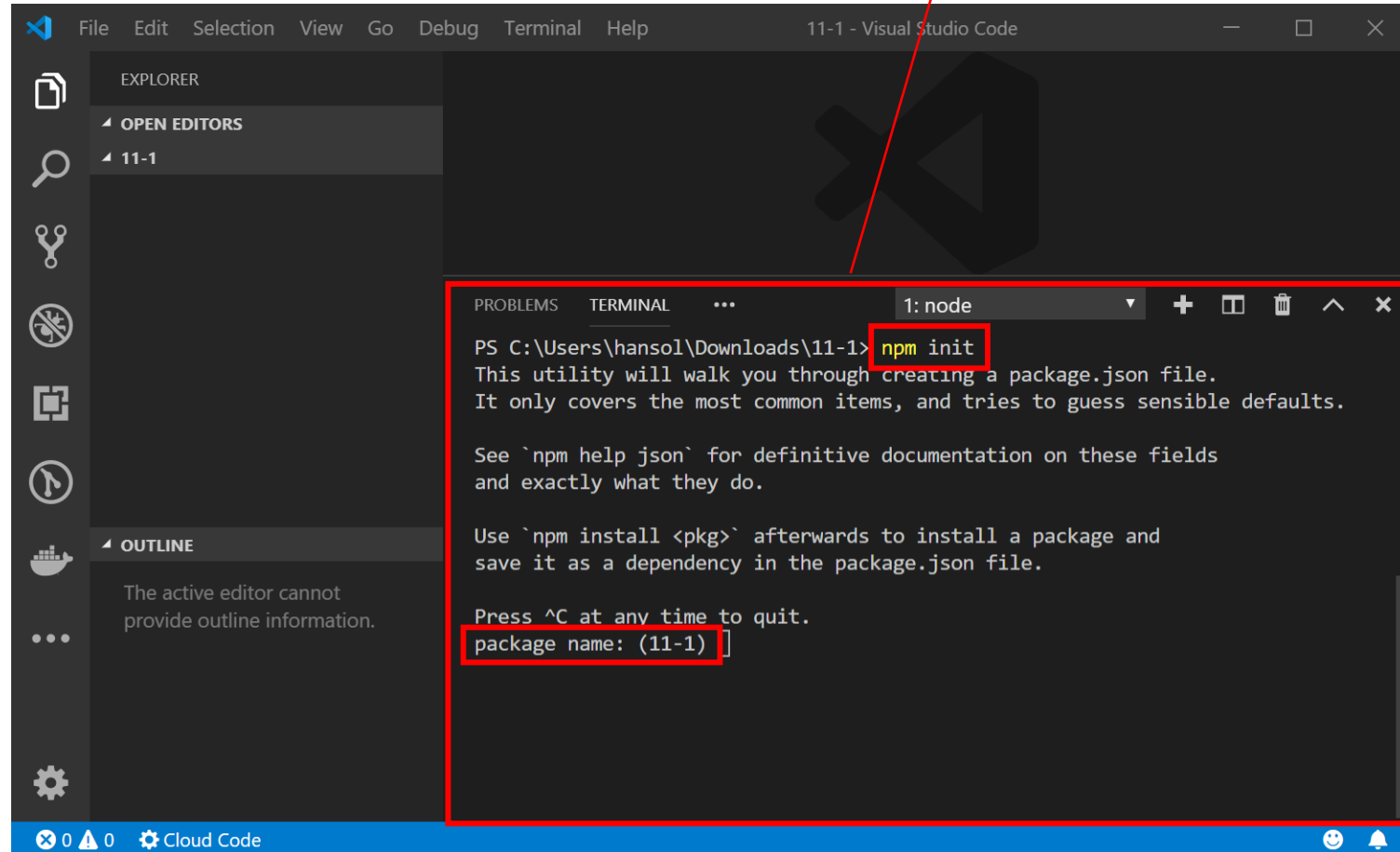
- NodeJS 모듈을 공유하고 다운받는 저장소
- CLI (명령줄 인터페이스)로 간편하게 모듈 관리
- 의존성 관리
 - A 모듈이 B 모듈을 사용할 때 A모듈은 B모듈에 의존성이 있다고 말함
 - package.json 파일로 프로젝트 의존성 관리

npm init 명령어

- NodeJS 프로젝트를 시작하기 위한 명령어
- package.json 파일을 생성
 - 프로젝트의 이름, 버전, 개발자 정보 등의 메타데이터
 - NPM 모듈의 의존성 정보 등을 담고 있음

npm init 명령어

Ct기 + ~
내장 터미널
열기



The screenshot shows the Visual Studio Code interface with a terminal window open at the bottom. The terminal title is '1: node'. The command prompt shows 'PS C:\Users\hansol\Downloads\11-1> npm init'. The output of the command is displayed below the prompt, explaining the utility and providing instructions. The 'package name: (11-1)' prompt is visible at the bottom of the terminal output.

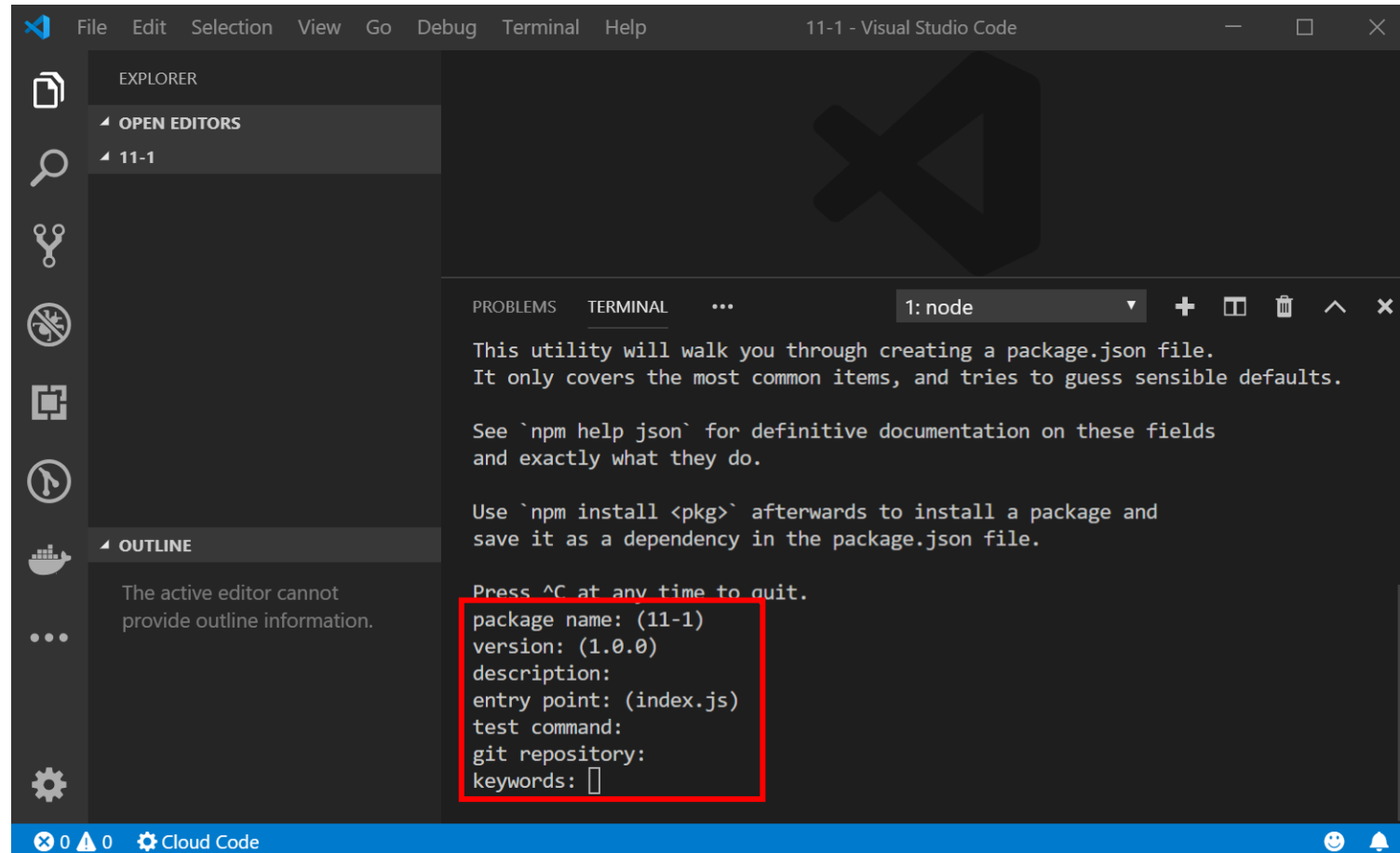
```
PS C:\Users\hansol\Downloads\11-1> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` 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: (11-1)
```

npm init 명령어

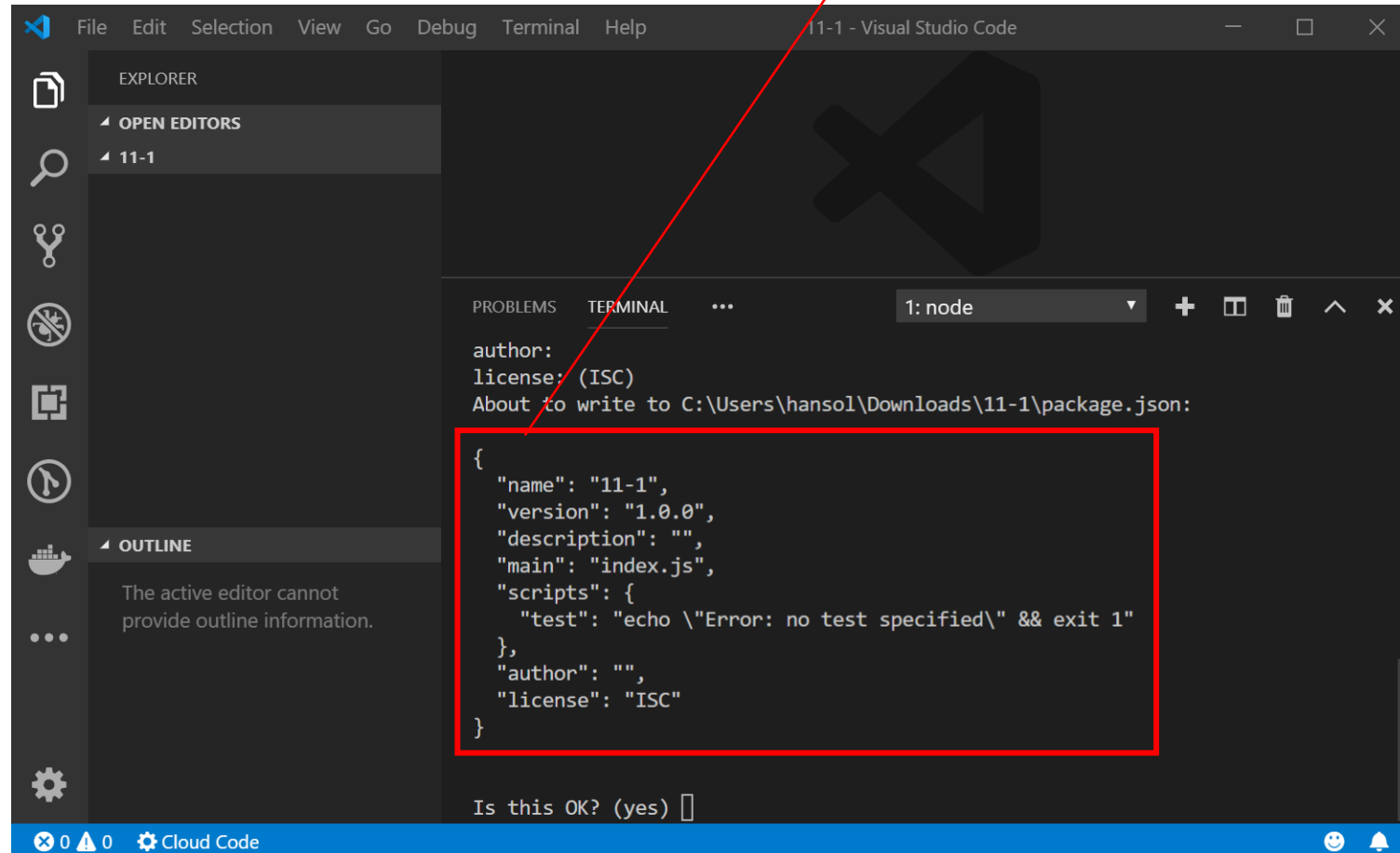


The screenshot shows the Visual Studio Code interface with the terminal panel open. The terminal output of the `npm init` command is displayed. A red box highlights the default values for the package name, version, entry point, and test command.

```
1: node  
  
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 json` 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: (11-1)  
version: (1.0.0)  
description:  
entry point: (index.js)  
test command:  
git repository:  
keywords: []
```

npm init 명령어

틀린 정보가 있다면
ctrl + c 입력 후 명령줄 프로그램
빠져 나와 다시 npm init 실행



```
File Edit Selection View Go Debug Terminal Help 11-1 - Visual Studio Code

EXPLORER
└─ OPEN EDITORS
  └─ 11-1

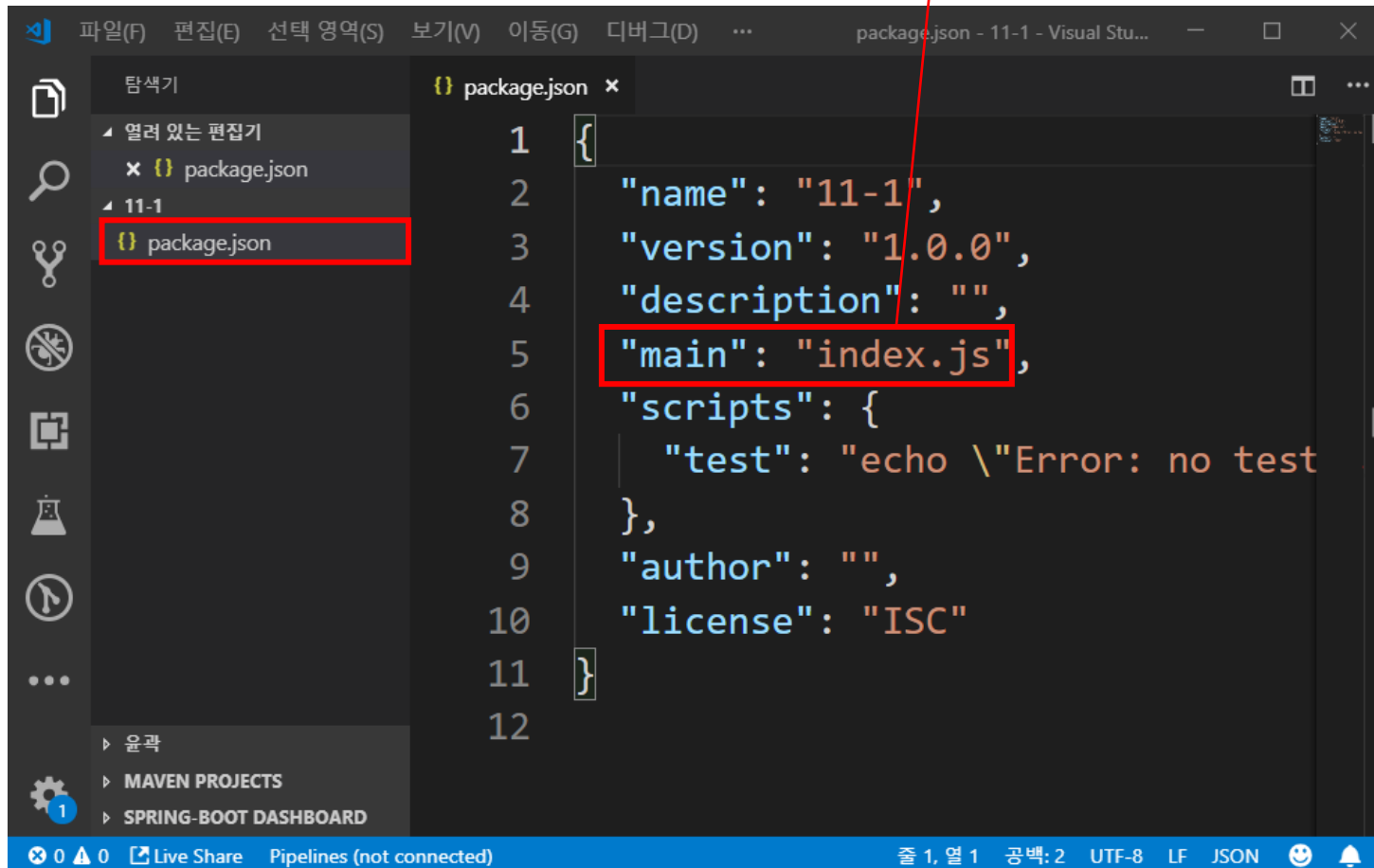
PROBLEMS TERMINAL ... 1: node
author:
license: (ISC)
About to write to C:\Users\hansol\Downloads\11-1\package.json:

{
  "name": "11-1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) [ ]
```

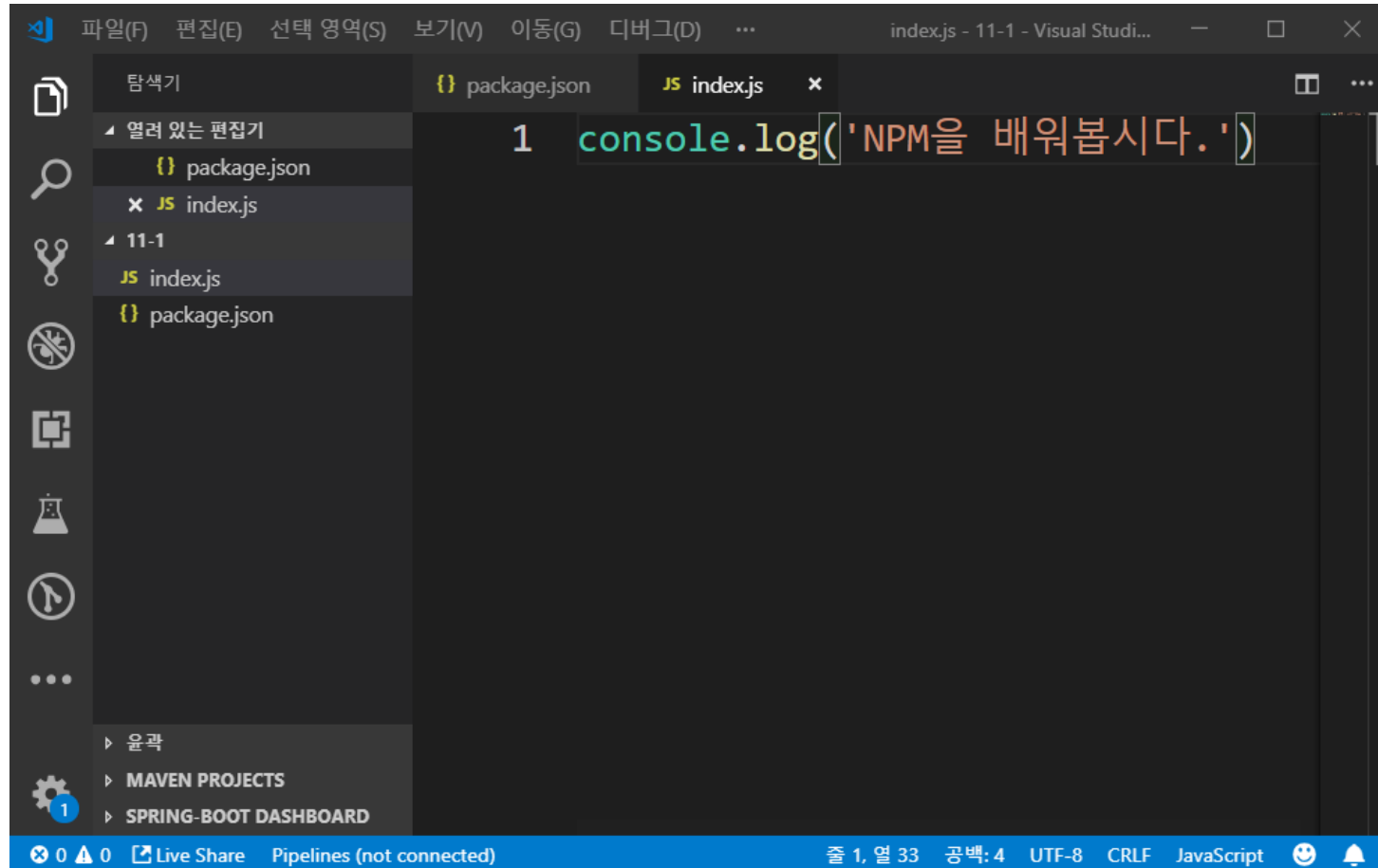
npm init 명령어

엔트리 포인트 - index.js



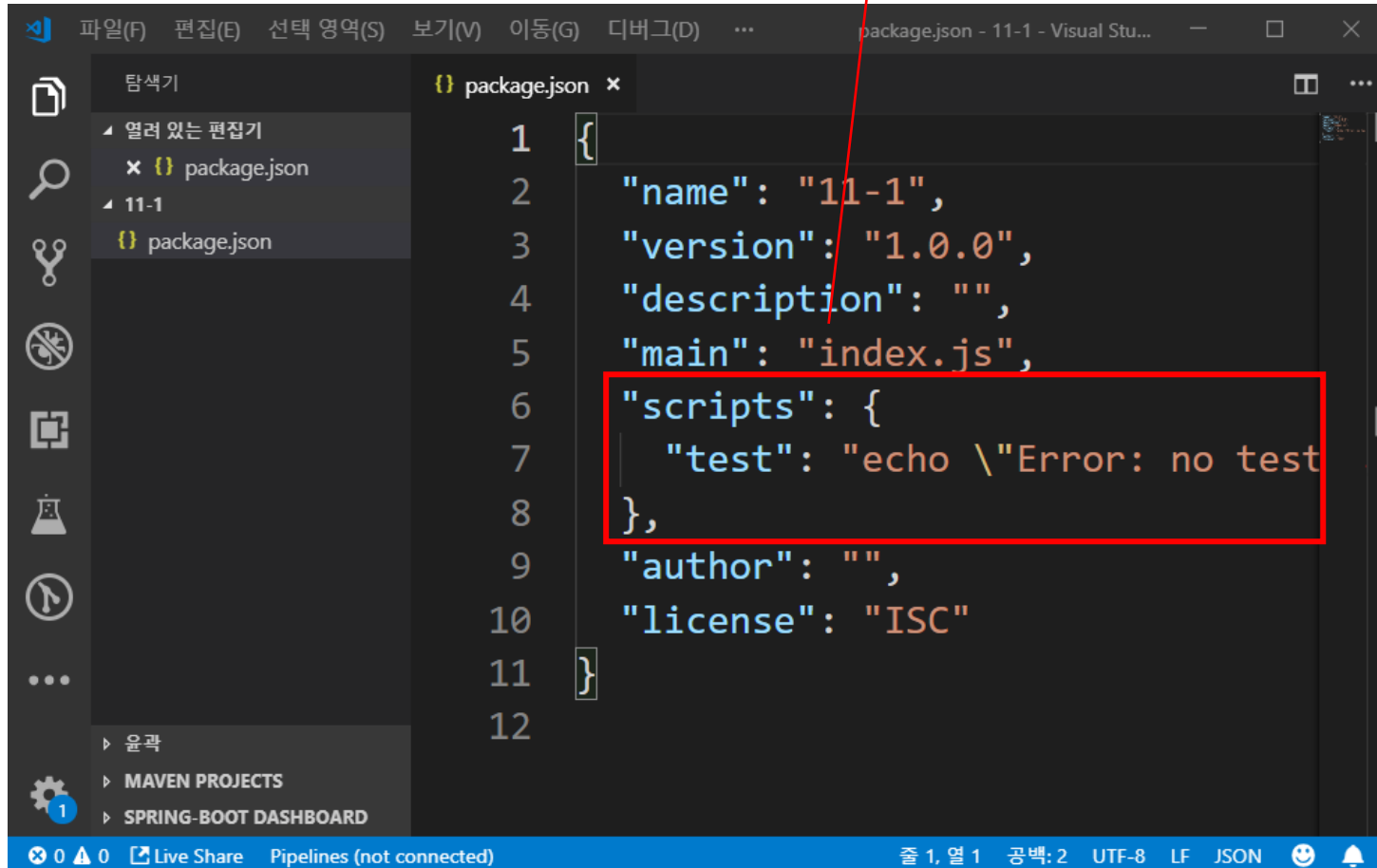
```
1 {
2   "name": "11-1",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test\"
8   },
9   "author": "",
10  "license": "ISC"
11 }
12
```


예제 코드 작성



npm run 명령어

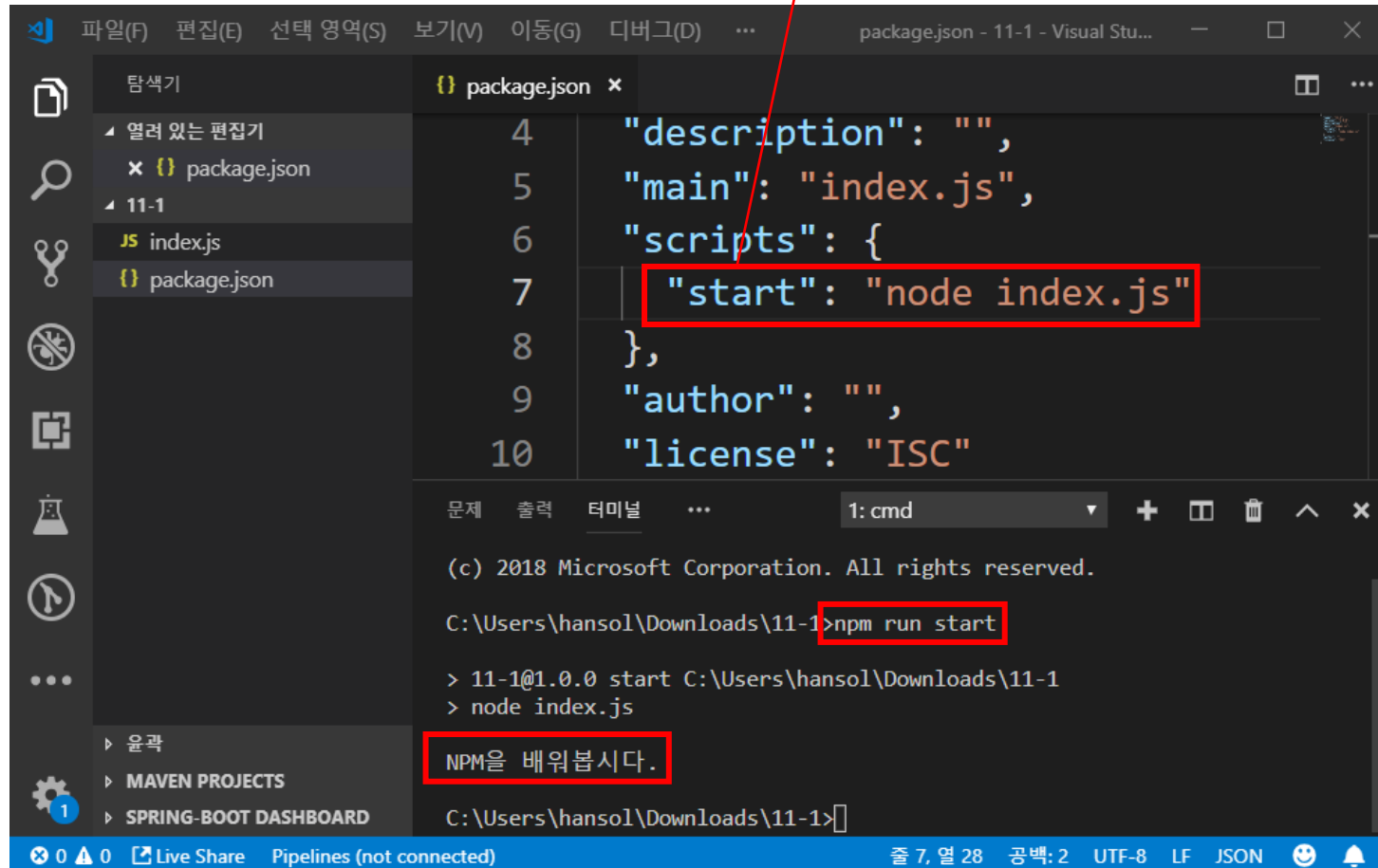
Scripts 프로퍼티에 적어 놓은 명령어들은
npm run 명령어이름
입력해서 실행 가능



```
1 {
2   "name": "11-1",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test\"",
8   },
9   "author": "",
10  "license": "ISC"
11 }
12
```

npm run 명령어

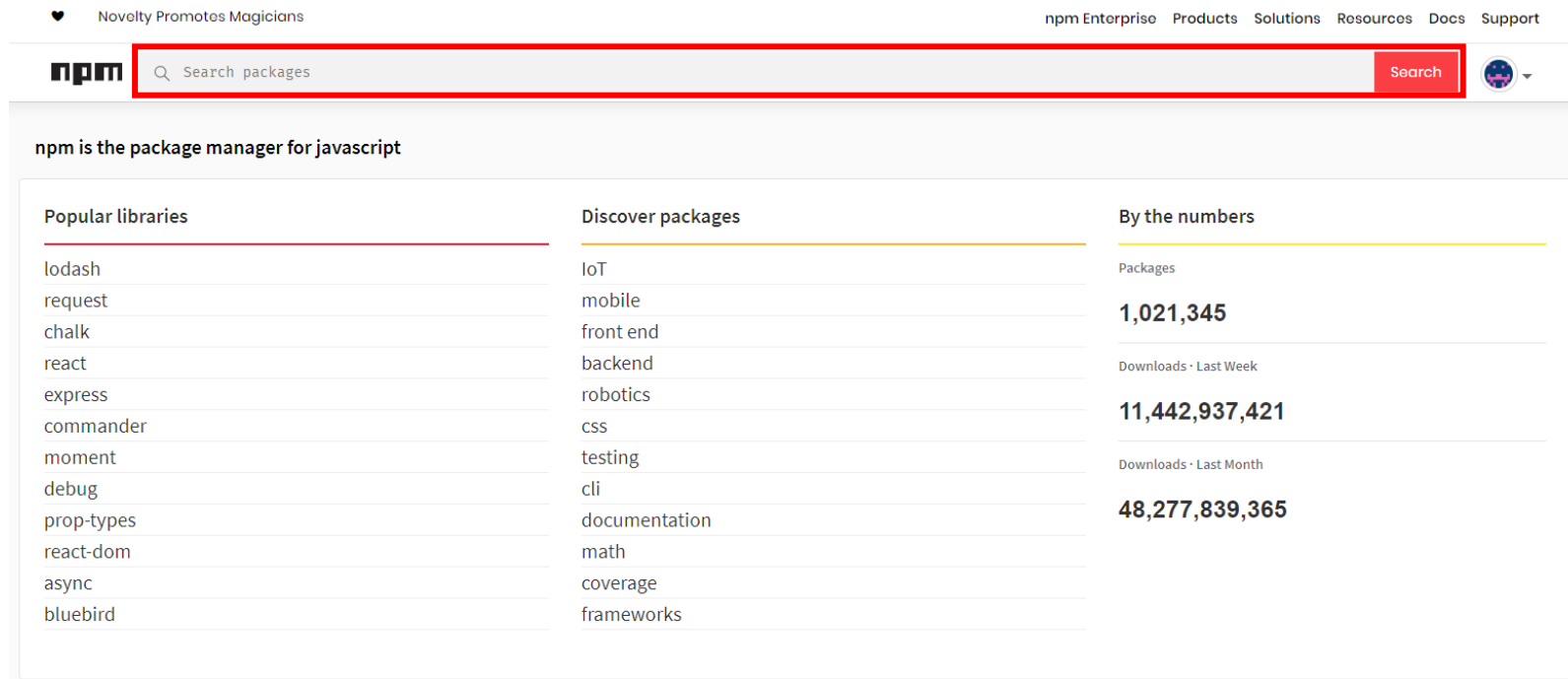
npm run start 명령입력
Node index.js 명령을 실행



```
package.json - 11-1 - Visual Stu...  
package.json  
4 "description": "",  
5 "main": "index.js",  
6 "scripts": {  
7   "start": "node index.js"  
8 },  
9 "author": "",  
10 "license": "ISC"  
  
문제 출력 터미널 ... 1: cmd  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\Users\hansol\Downloads\11-1>npm run start  
  
> 11-1@1.0.0 start C:\Users\hansol\Downloads\11-1  
> node index.js  
  
NPM을 배워봅시다.  
C:\Users\hansol\Downloads\11-1>
```

NPM 패키지 검색하기

- <https://www.npmjs.com/>



The screenshot shows the npm website homepage. At the top, there is a navigation bar with links for 'npm Enterprise', 'Products', 'Solutions', 'Resources', 'Docs', and 'Support'. Below this is a search bar with the text 'Search packages' and a red 'Search' button. The main content area is titled 'npm is the package manager for javascript'. It is divided into three columns: 'Popular libraries', 'Discover packages', and 'By the numbers'. The 'Popular libraries' column lists various packages like lodash, request, chalk, react, express, commander, moment, debug, prop-types, react-dom, async, and bluebird. The 'Discover packages' column lists categories like IoT, mobile, front end, backend, robotics, css, testing, cli, documentation, math, coverage, and frameworks. The 'By the numbers' column displays statistics: 1,021,345 Packages, Downloads - Last Week: 11,442,937,421, and Downloads - Last Month: 48,277,839,365.

Novelty Promotes Magicians

npm Enterprise Products Solutions Resources Docs Support

npm Search packages Search

npm is the package manager for javascript

| Popular libraries | Discover packages | By the numbers |
|-------------------|-------------------|------------------------|
| lodash | IoT | Packages |
| request | mobile | 1,021,345 |
| chalk | front end | Downloads - Last Week |
| react | backend | 11,442,937,421 |
| express | robotics | Downloads - Last Month |
| commander | css | 48,277,839,365 |
| moment | testing | |
| debug | cli | |
| prop-types | documentation | |
| react-dom | math | |
| async | coverage | |
| bluebird | frameworks | |

NPM 패키지 검색하기

npm

17659 packages found

Sort Packages

Optimal

Popularity

Quality


Maintenance

Who's Hiring?

express exact match

Fast, unopinionated, minimalist web framework

express framework sinatra web rest restful router app api

 **dougwilson** published 4.17.1 • a month ago

path-to-regexp

Express style path to RegExp utility

express regexp route routing

NPM 패키지 검색하기

express

4.17.1 • Public • Published a month ago

Readme

30 Dependencies

34,248 Dependents

263 Versions

express

Fast, unopinionated, minimalist web framework for **node**.

npm v4.17.1 downloads 40M/month linux passing windows passing coverage 34%

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

app.get('/', function (req, res) {
```

install

```
> npm i express
```

↓ weekly downloads

9,412,888



version

4.17.1

license

MIT

open issues

114

pull requests

53

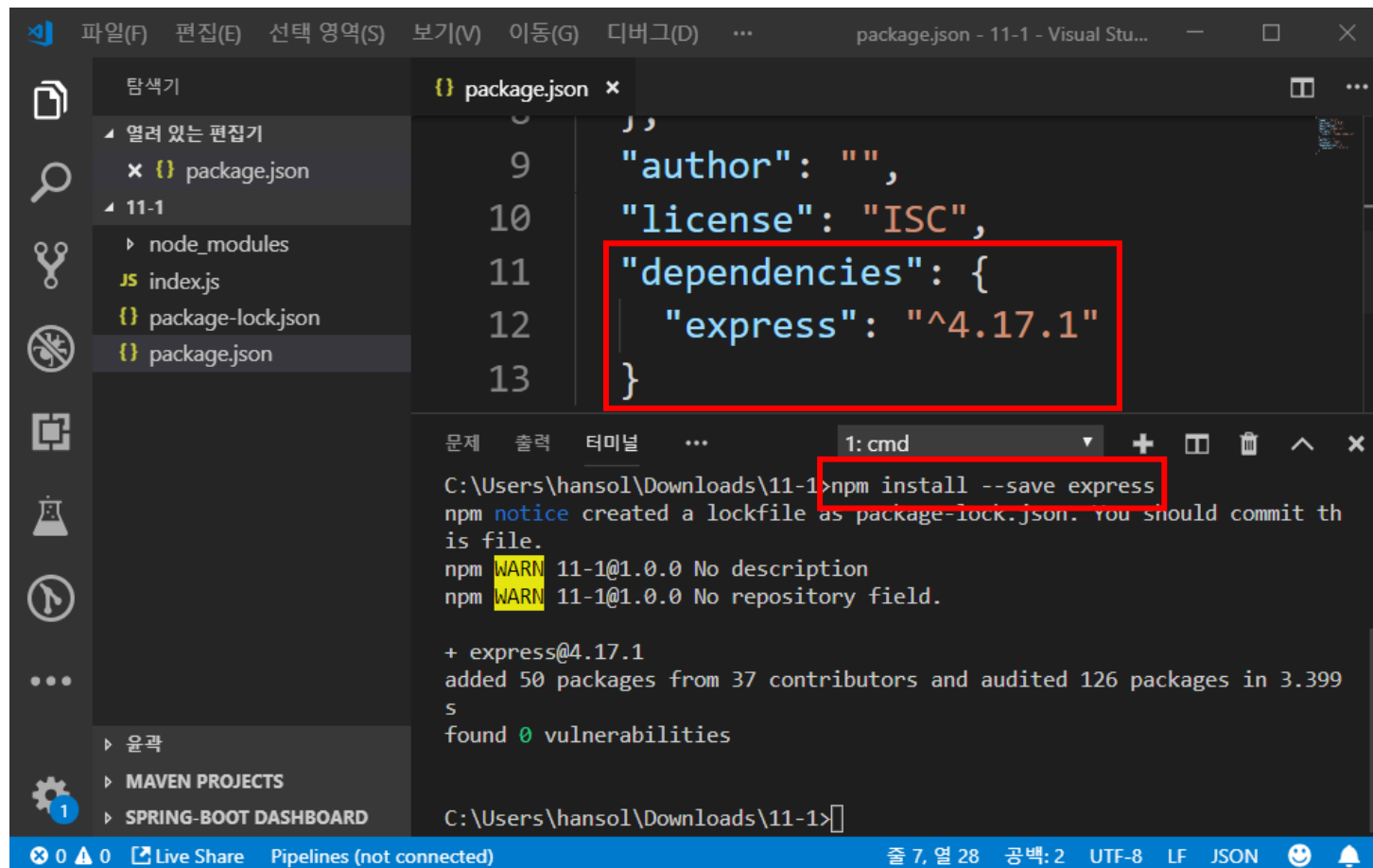
homepage

repository

npm install 명령어

- npm install {모듈이름}
 - 예) express 모듈 설치
npm install express
- ./node_modules 폴더에 모듈을 설치
- package.json 에 의존성 정보를 저장하려면 --save 옵션과 함께 실행

npm install 명령어



The screenshot shows the Visual Studio Code interface with a project named '11-1'. The file explorer on the left shows the project structure, including 'package.json', 'package-lock.json', and 'index.js'. The main editor displays the 'package.json' file, where the 'dependencies' section is highlighted with a red box, showing the installation of 'express' version '^4.17.1'. The terminal at the bottom shows the command 'npm install --save express' being executed, followed by npm's output, including a notice about the lockfile and warnings about missing description and repository fields. The status bar at the bottom indicates the current file is 'package.json' and the encoding is 'UTF-8'.

```
package.json
{
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1"
  }
}
```

```
C:\Users\hansol\Downloads\11-1>npm install --save express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN 11-1@1.0.0 No description
npm WARN 11-1@1.0.0 No repository field.

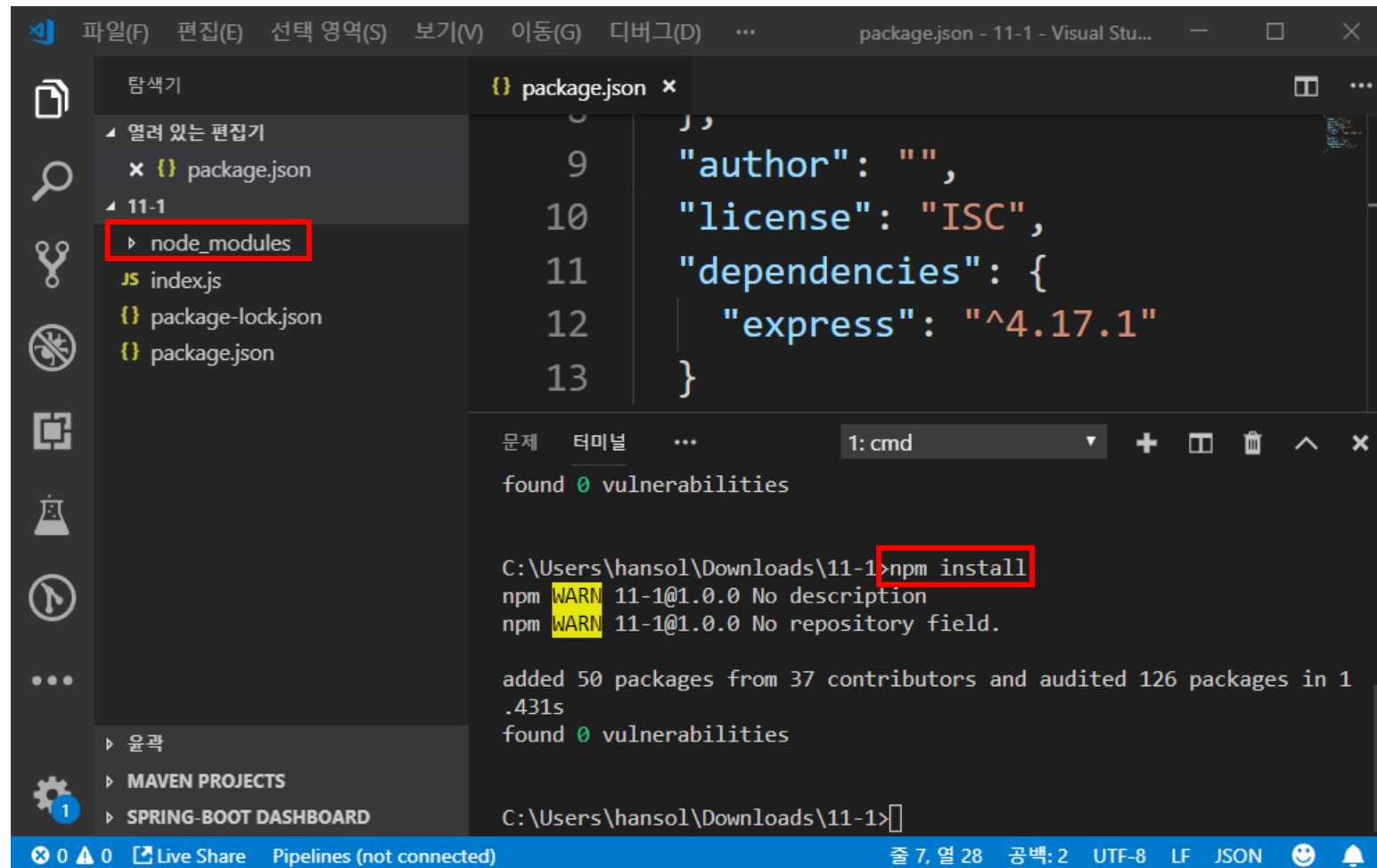
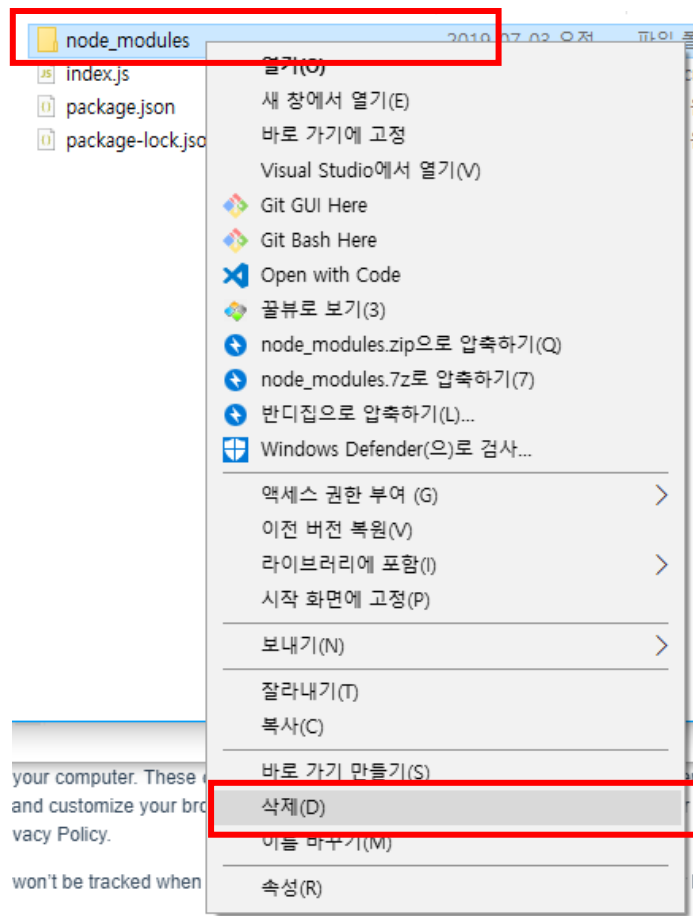
+ express@4.17.1
added 50 packages from 37 contributors and audited 126 packages in 3.399s
found 0 vulnerabilities

C:\Users\hansol\Downloads\11-1>
```


npm install 명령어

| 11-1 > node_modules | | | | node_modules 검색 |
|---------------------|------------------|-------|----|-----------------|
| 이름 | 수정한 날짜 | 유형 | 크기 | |
| .bin | 2019-07-03 오전... | 파일 폴더 | | |
| accepts | 2019-07-03 오전... | 파일 폴더 | | |
| array-flatten | 2019-07-03 오전... | 파일 폴더 | | |
| body-parser | 2019-07-03 오전... | 파일 폴더 | | |
| bytes | 2019-07-03 오전... | 파일 폴더 | | |
| content-disposition | 2019-07-03 오전... | 파일 폴더 | | |
| content-type | 2019-07-03 오전... | 파일 폴더 | | |
| cookie | 2019-07-03 오전... | 파일 폴더 | | |
| cookie-signature | 2019-07-03 오전... | 파일 폴더 | | |
| debug | 2019-07-03 오전... | 파일 폴더 | | |
| depd | 2019-07-03 오전... | 파일 폴더 | | |
| destroy | 2019-07-03 오전... | 파일 폴더 | | |
| ee-first | 2019-07-03 오전... | 파일 폴더 | | |
| encodeurl | 2019-07-03 오전... | 파일 폴더 | | |
| escape-html | 2019-07-03 오전... | 파일 폴더 | | |
| etag | 2019-07-03 오전... | 파일 폴더 | | |
| express | 2019-07-03 오전... | 파일 폴더 | | |
| finalhandler | 2019-07-03 오전... | 파일 폴더 | | |
| forwarded | 2019-07-03 오전... | 파일 폴더 | | |
| fresh | 2019-07-03 오전... | 파일 폴더 | | |

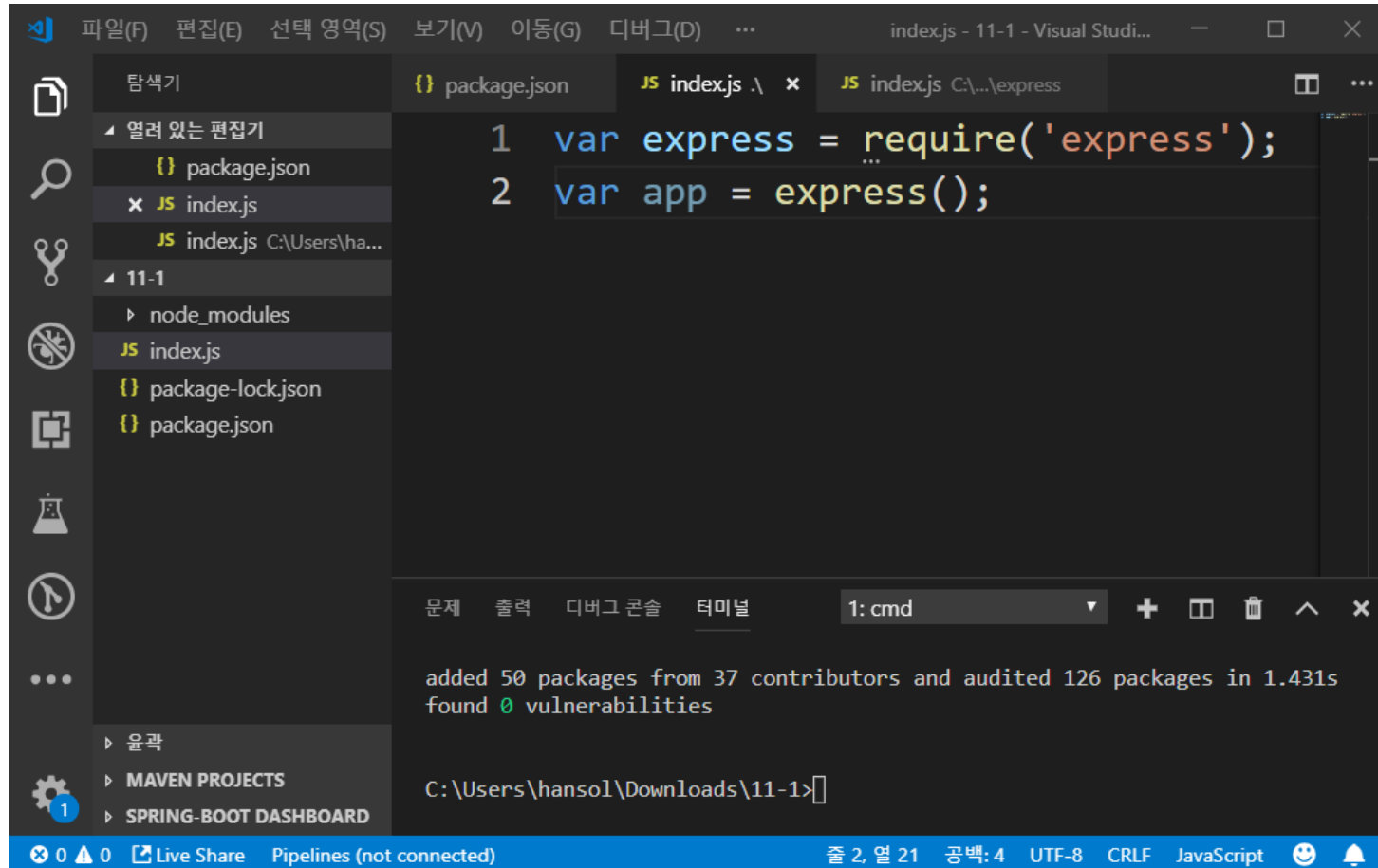
npm install 명령어



require 함수로 외부 모듈 이용하기

- require 함수를 모듈명을 인자로 실행
- 파일명을 전달할 때와 마찬가지로 해당 모듈을 반환

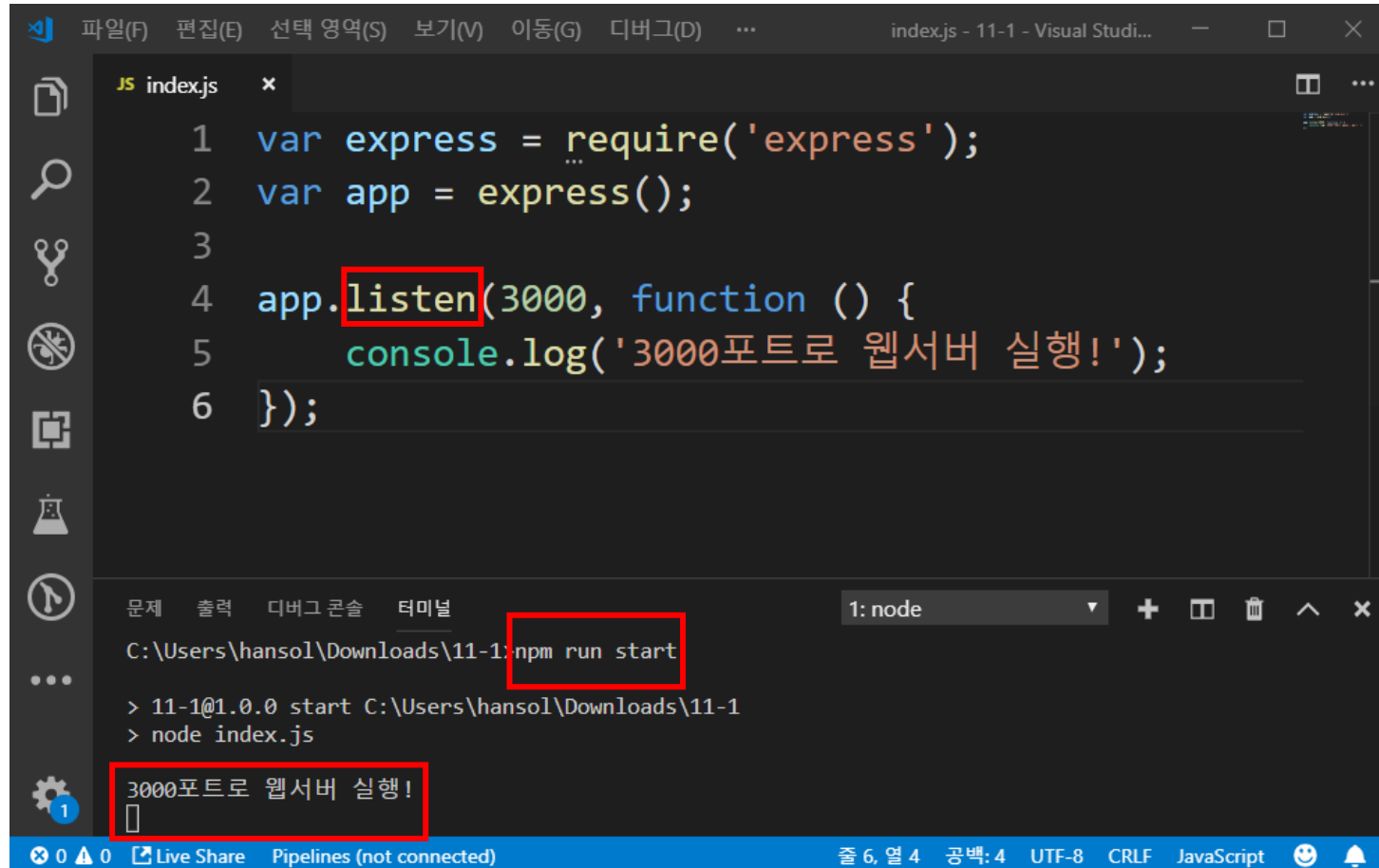
require 함수로 외부 모듈 이용하기



Express

- 고성능 웹 프레임워크
 - 프로그래머가 작성한 함수를 실행해 HTTP요청에 대한 HTTP응답을 프로그래머가 원하는 대로 만들어 HTTP 클라이언트에게 응답할 수 있음
 - 요청한 URL이나 HTTP 메서드에 따라 원하는 함수를 실행시켜 주는 **라우팅** 기능
 - HTML 의 내용 일부분을 변수에 저장된 값으로 치환해 동적인 HTML 을 만들어 응답하는 **뷰 템플릿 엔진**도 가지고 있음
 - 미들웨어로 기능 확장

Express 앱으로 HTTP 요청 처리하기



The image shows a Visual Studio Code editor window with a file named `index.js`. The code defines an Express application that listens on port 3000 and logs a message to the console. The `listen` method is highlighted with a red box. Below the code editor is a terminal window with the command `npm run start` highlighted by a red box. The terminal output shows the command being executed and the console log message `3000포트로 웹서버 실행!` being displayed, which is also highlighted by a red box. The status bar at the bottom indicates the file encoding is UTF-8 and the line ending is CRLF.

```
1 var express = require('express');
2 var app = express();
3
4 app.listen(3000, function () {
5     console.log('3000포트로 웹서버 실행!');
6 });
```

문제 출력 디버그 콘솔 터미널

1: node

C:\Users\hansol\Downloads\11-1> npm run start

> 11-1@1.0.0 start C:\Users\hansol\Downloads\11-1
> node index.js

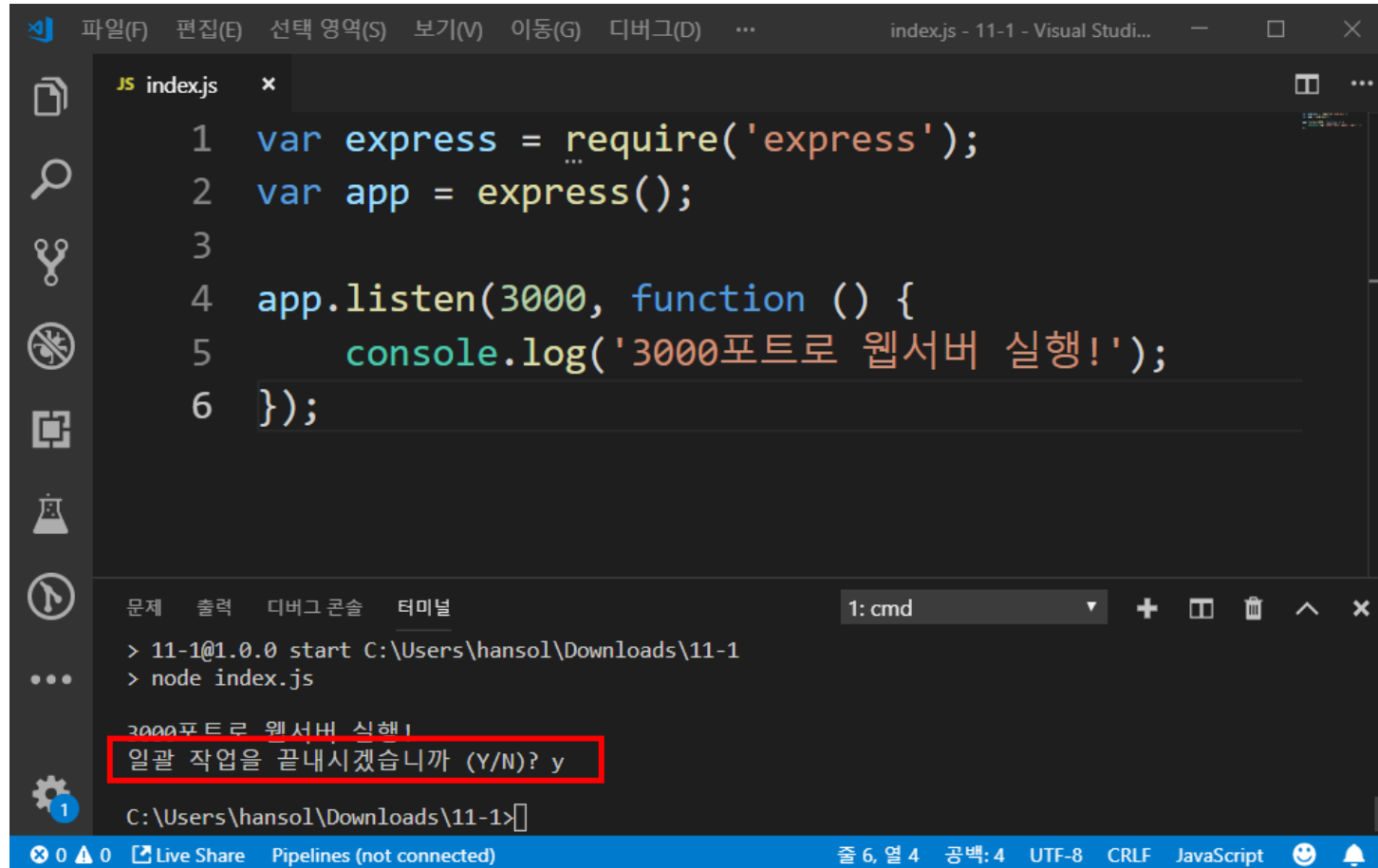
3000포트로 웹서버 실행!
█

0 0 Live Share Pipelines (not connected) 줄 6, 열 4 공백: 4 UTF-8 CRLF JavaScript

Express 앱으로 HTTP 요청 처리하기



Express 앱으로 HTTP 요청 처리하기



The image shows a Visual Studio Code window with a file named `index.js` open. The code defines an Express application that listens on port 3000 and logs a message to the console. Below the code editor, the integrated terminal shows the command to run the application, and the output confirms it is running on port 3000. A red box highlights the prompt to press any key to exit.

```
1 var express = require('express');
2 var app = express();
3
4 app.listen(3000, function () {
5     console.log('3000포트로 웹서버 실행!');
6 });
```

문제 출력 디버그 콘솔 터미널

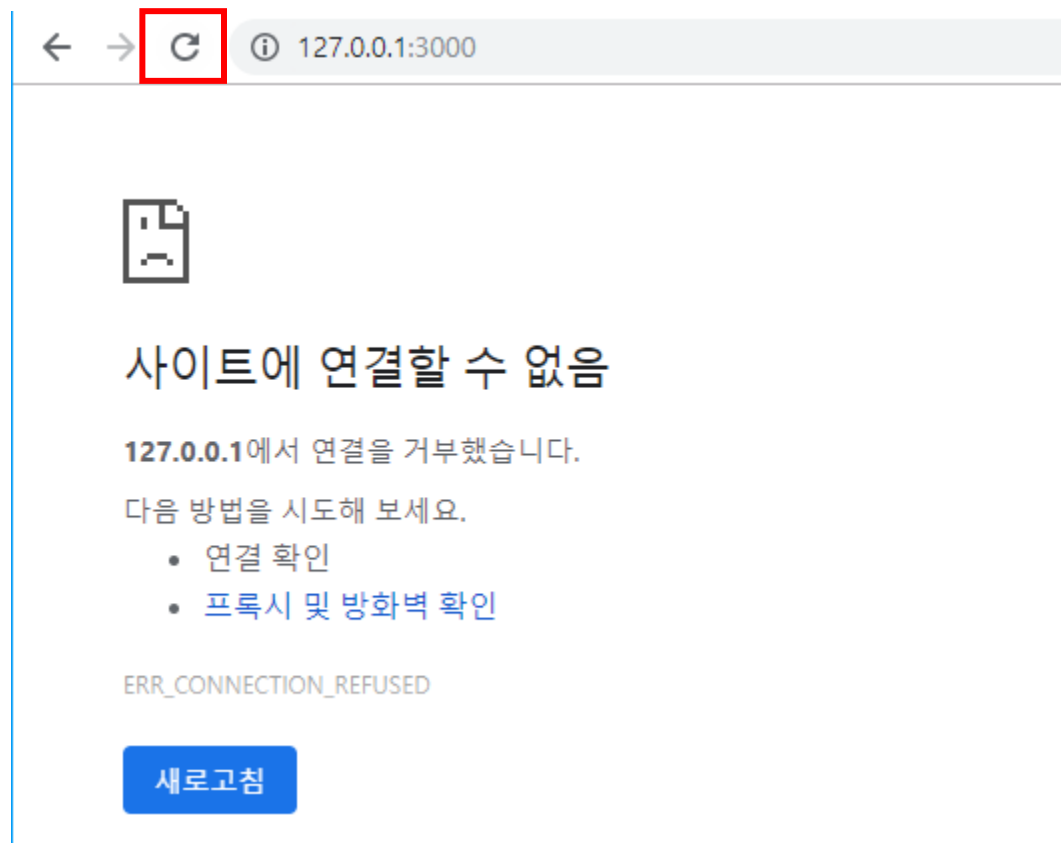
1: cmd

```
> 11-1@1.0.0 start C:\Users\hansol\Downloads\11-1
> node index.js

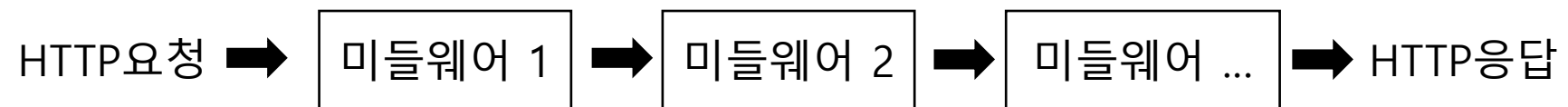
3000포트로 웹서버 실행!
일괄 작업을 끝내시겠습니까 (Y/N)? y
C:\Users\hansol\Downloads\11-1>
```

0 0 Live Share Pipelines (not connected) 줄 6, 열 4 공백: 4 UTF-8 CRLF JavaScript

Express 앱으로 HTTP 요청 처리하기



Express



미들웨어 콜백함수의 인수

- Request 객체
 - HTTP 요청에 대한 정보가 담겨 있음

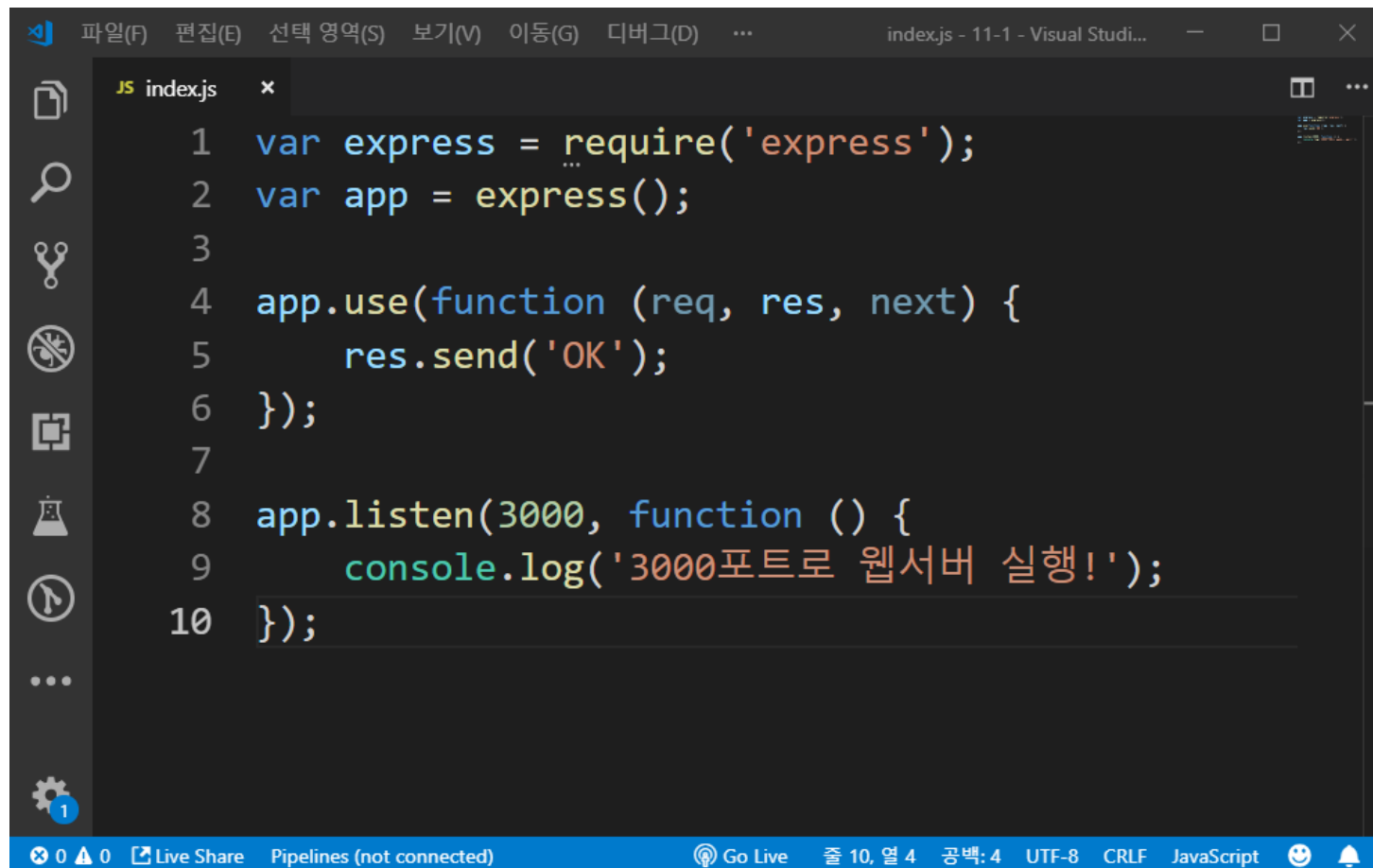
미들웨어 콜백함수의 인수

- Request 객체
 - HTTP 요청에 대한 정보가 담겨 있음
- Response 객체
 - 브라우저에 전송할 HTTP 응답을 어떻게 만들지에 대한 정보를 설정

미들웨어 콜백함수의 인수

- Request 객체
 - HTTP 요청에 대한 정보가 담겨 있음
- Response 객체
 - 브라우저에 전송할 HTTP 응답을 어떻게 만들지에 대한 정보를 설정
- next 함수
 - 이전 미들웨어에서 다음 미들웨어도 실행해야 한다고 알려주는 함수

Express 미들웨어



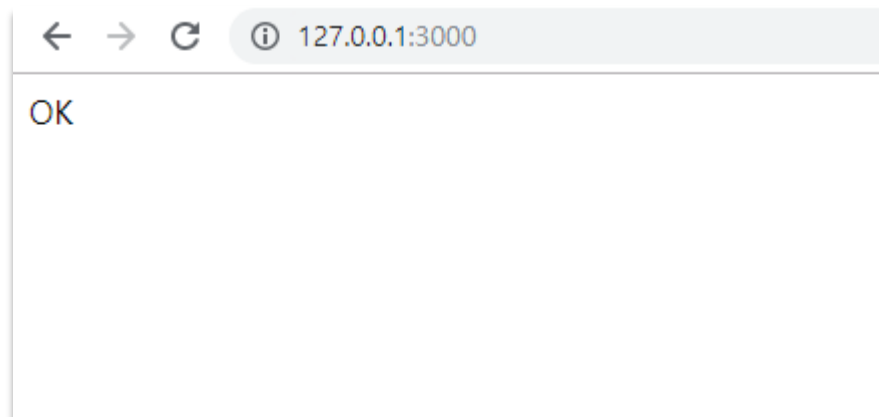
The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', and '디버그(D)'. The title bar indicates the file is 'index.js' in a workspace named '11-1 - Visual Studi...'. The left sidebar contains icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The main editor window displays the following JavaScript code in 'index.js':

```
1 var express = require('express');
2 var app = express();
3
4 app.use(function (req, res, next) {
5     res.send('OK');
6 });
7
8 app.listen(3000, function () {
9     console.log('3000포트로 웹서버 실행!');
10 });
```

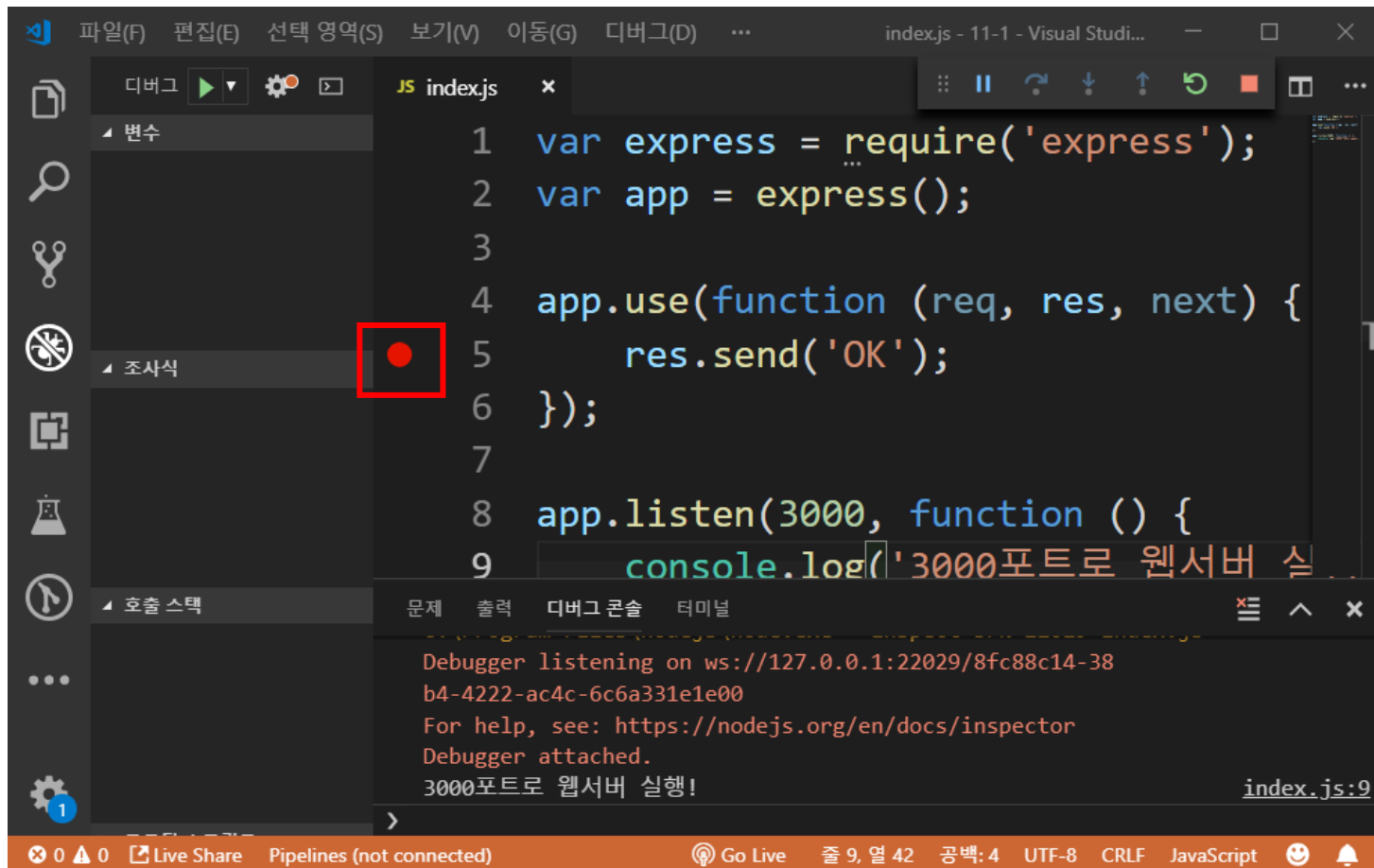
The bottom status bar shows '0 errors, 0 warnings', 'Live Share', 'Pipelines (not connected)', 'Go Live', '줄 10, 열 4', '공백: 4', 'UTF-8', 'CRLF', 'JavaScript', and a notification bell.

Express 미들웨어

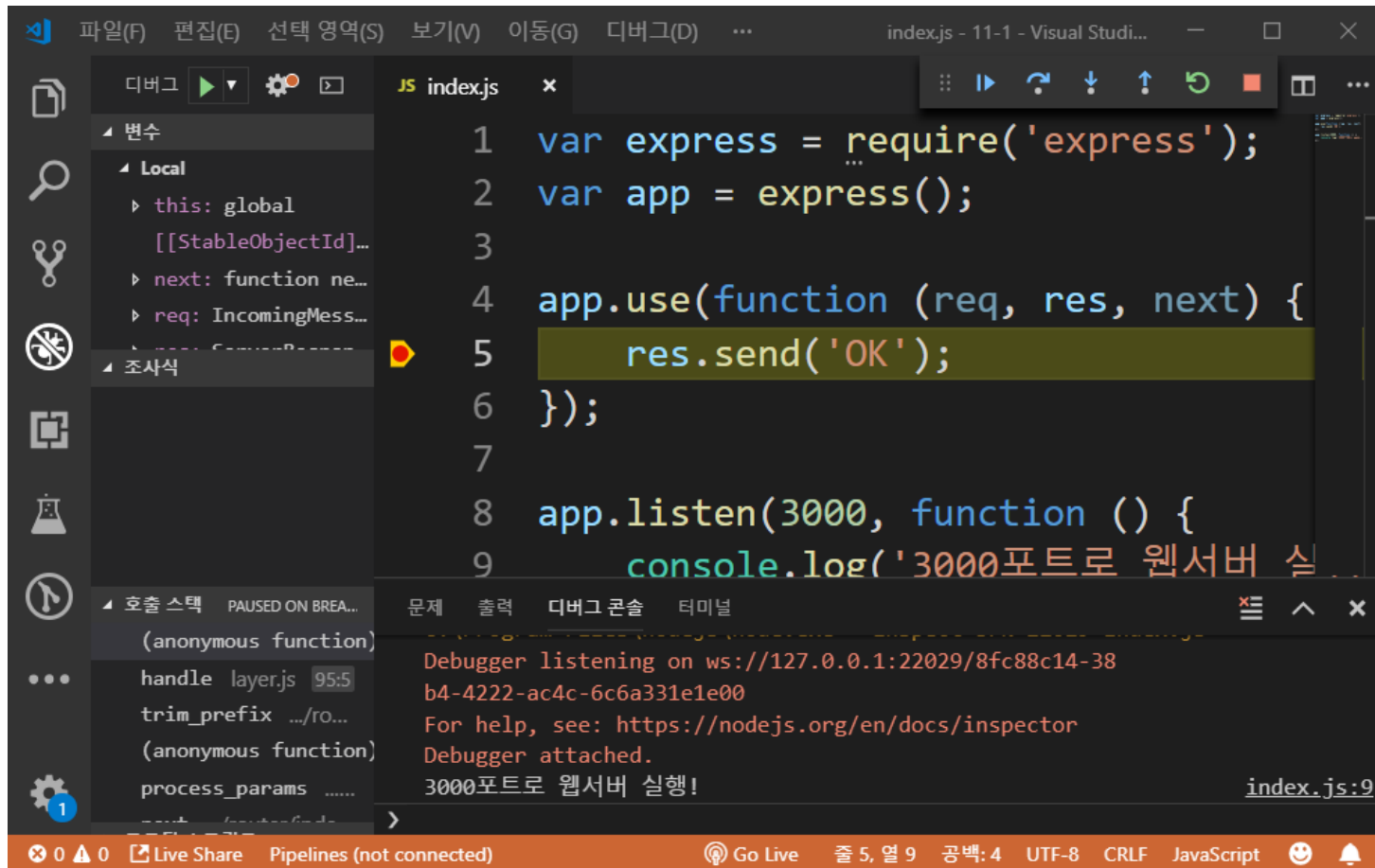
```
문제  출력  디버그 콘솔  터미널
Debugger listening on ws://127.0.0.1:22029/8fc88c14-38
b4-4222-ac4c-6c6a331e1e00
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
3000포트로 웹서버 실행! index.js:9
>
```



Express 미들웨어



Express 미들웨어

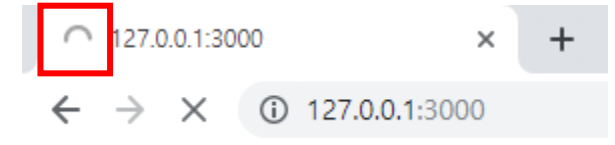


The screenshot shows the Visual Studio Code editor with a JavaScript file named `index.js`. The code defines an Express application that listens on port 3000 and responds with 'OK'.

```
1 var express = require('express');
2 var app = express();
3
4 app.use(function (req, res, next) {
5   res.send('OK');
6 });
7
8 app.listen(3000, function () {
9   console.log('3000포트로 웹서버 실행!');
10});
```

The left sidebar shows the '변수' (Variables) pane with a 'Local' scope containing variables like `this`, `next`, and `req`. The bottom pane shows the '디버그 콘솔' (Debug Console) with the following output:

```
Debugger listening on ws://127.0.0.1:22029/8fc88c14-38b4-4222-ac4c-6c6a331e1e00
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
3000포트로 웹서버 실행! index.js:9
```



요약

- NPM 소개
- NPM로 외부 모듈 설치
- Express.js 모듈 설치 및 사용

차시 예고

- 11-2 : **RESTful API 알아보기**
 - RESTful API 소개
 - Express.js 로 RESTful API 만들기

강의를 마치겠습니다
수고하셨습니다

11주차_01 npm과 express모듈 알아보기