

Node.js

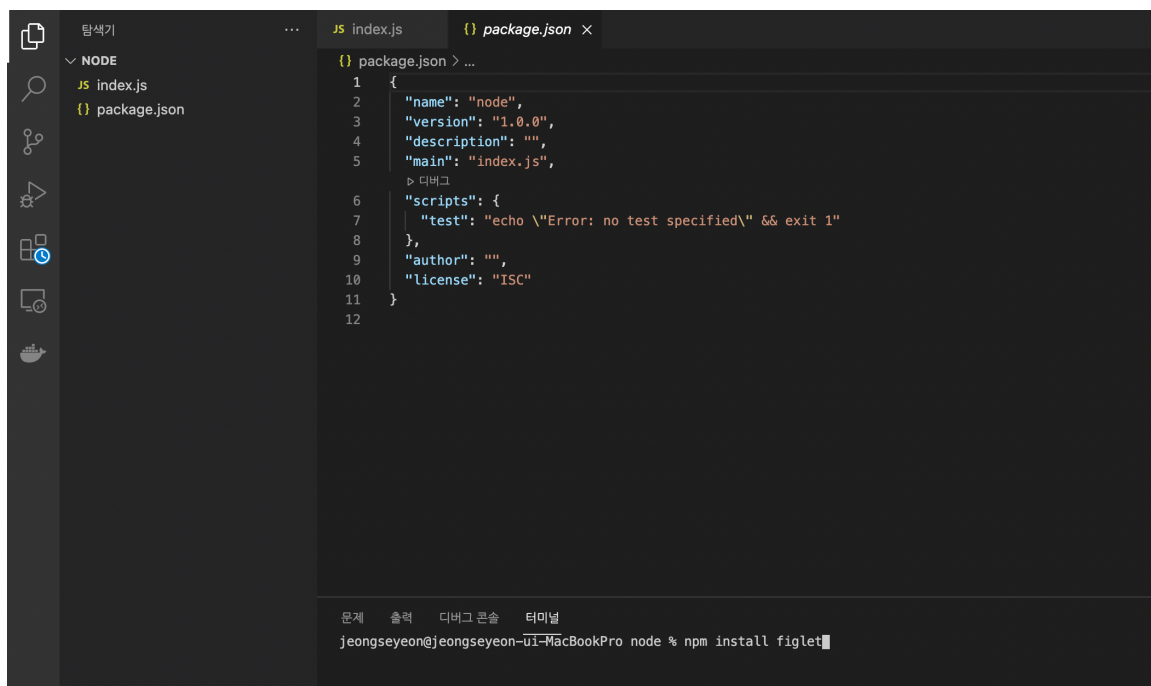
🕒 생성일	@2023년 1월 11일 오후 10:03
🕒 최종 편집 일시	@2023년 1월 11일 오후 11:35

npm init

- 설치하고 나면 package.js 가 생김

npm install figlet

- npm install figlet

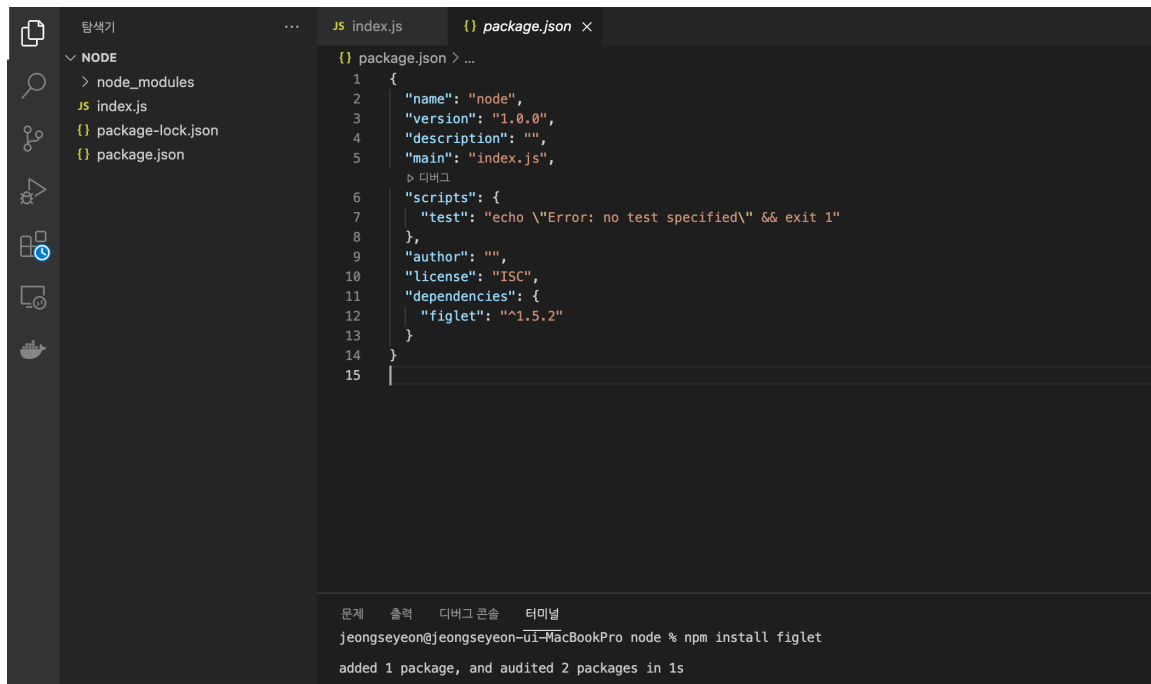


The screenshot shows a code editor with a sidebar on the left containing a file explorer. Under the 'NODE' folder, there are two files: 'index.js' and 'package.json'. The main editor area displays the content of 'package.json', which is a JSON object with the following structure:

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

At the bottom of the editor, there is a terminal window. The terminal shows the command 'npm install figlet' being executed in a shell. The prompt is 'jeongseyeon@jeongseyeon-U1-MacBookPro node %'.

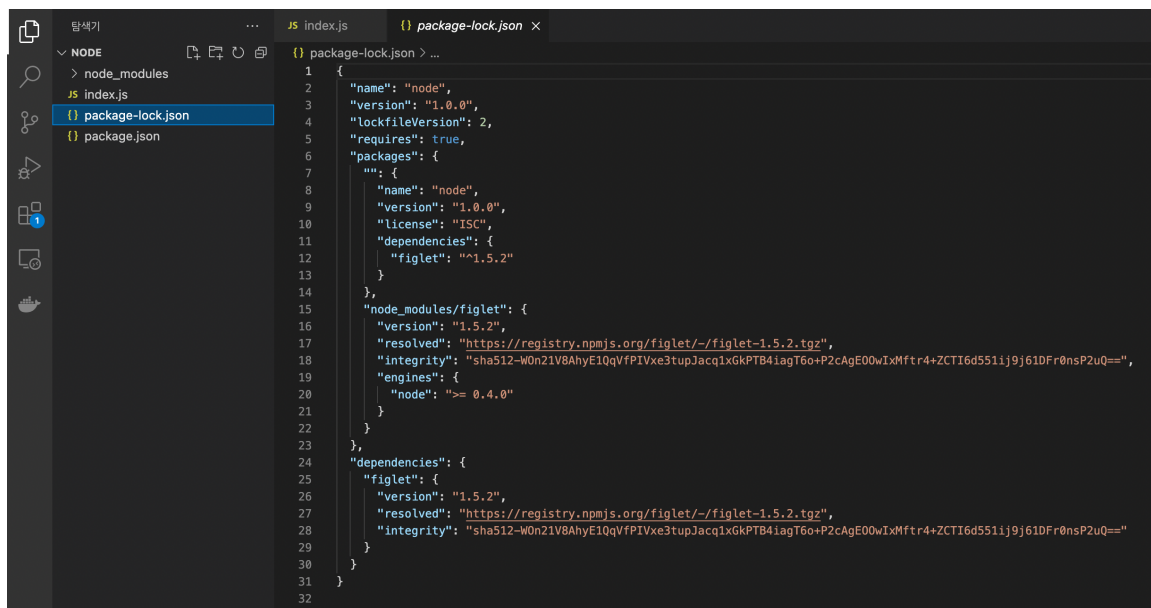
- npm install figlet 을 입력하면 package-lock.json이 생김



The screenshot shows the VS Code editor with the `package.json` file open. The file contains the following JSON:

```
{
  "name": "node",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "figlet": "^1.5.2"
  }
}
```

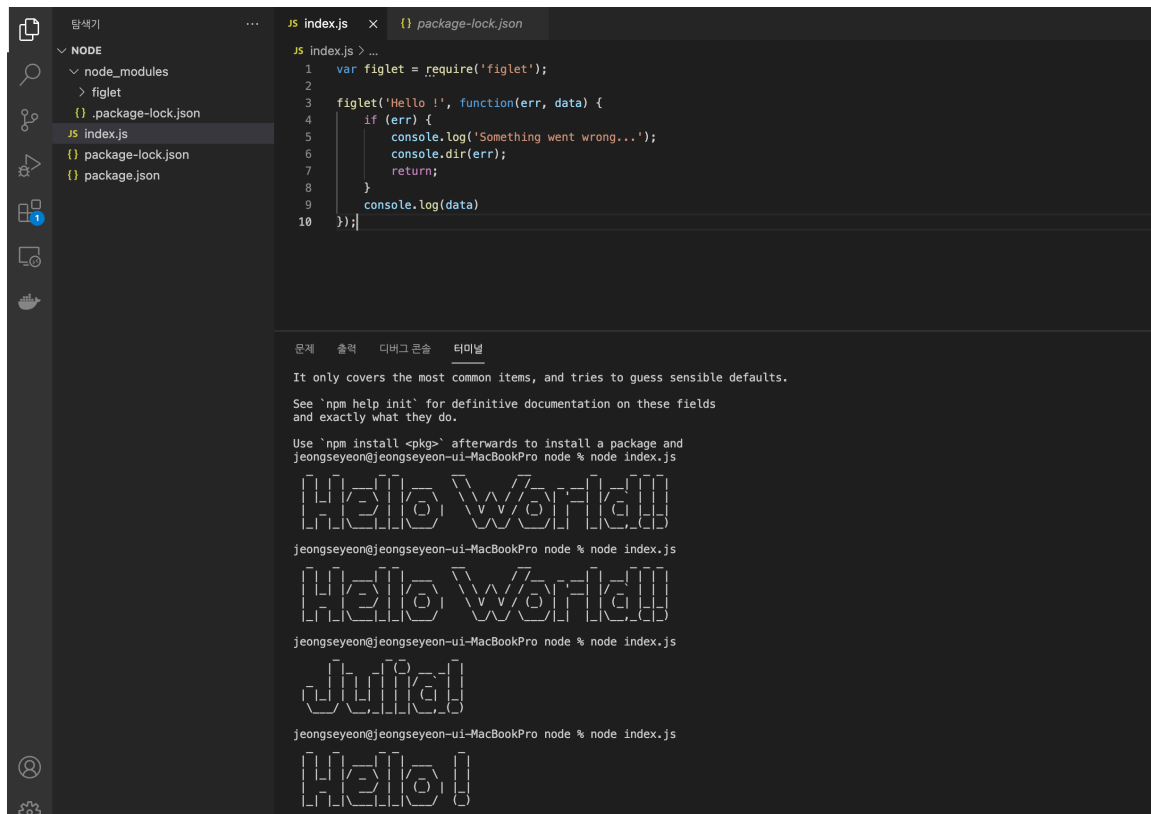
The terminal at the bottom shows the command `npm install figlet` being executed, with the output: `added 1 package, and audited 2 packages in 1s`.



The screenshot shows the VS Code editor with the `package-lock.json` file open. The file contains the following JSON:

```
{
  "name": "node",
  "version": "1.0.0",
  "lockfileVersion": 2,
  "requires": true,
  "packages": {
    "": {
      "name": "node",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "figlet": "^1.5.2"
      }
    },
    "node_modules/figlet": {
      "version": "1.5.2",
      "resolved": "https://registry.npmjs.org/figlet/-/figlet-1.5.2.tgz",
      "integrity": "sha512-W0n21VB8AhY1QqVfPIVxe3tupJacq1xGkPTB4iagT6o+P2cAgE00wIxMftr4+ZCTI6d551ij9j61DFr0nsP2uQ==",
      "engines": {
        "node": ">= 0.4.0"
      }
    }
  },
  "dependencies": {
    "figlet": {
      "version": "1.5.2",
      "resolved": "https://registry.npmjs.org/figlet/-/figlet-1.5.2.tgz",
      "integrity": "sha512-W0n21VB8AhY1QqVfPIVxe3tupJacq1xGkPTB4iagT6o+P2cAgE00wIxMftr4+ZCTI6d551ij9j61DFr0nsP2uQ=="
    }
  }
}
```

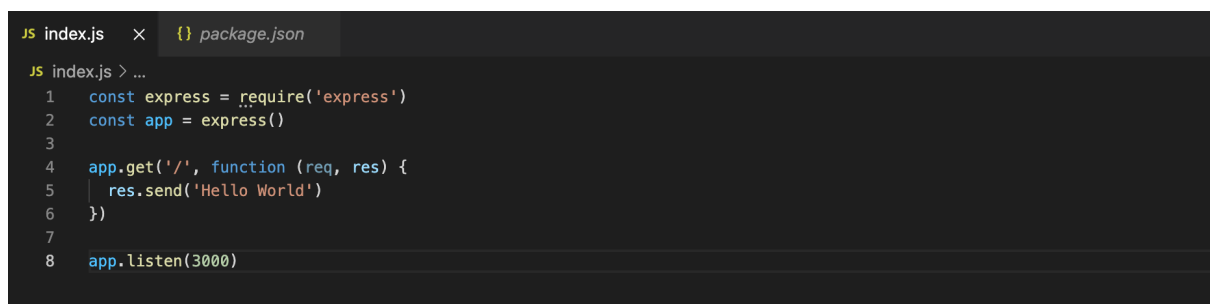
- package.json 과 package-lock.json 의 차이
 - package.json : 내용을 대략적으로 확인하기 위함
 - package-lock.json : 내용을 상세하게 확인하기 위함
- node_modules도 생김
- 애플리케이션 사용 결과



npm uninstall figlet

express

- web package



- listen(port)
 - port 란?

- 들어올 수 있는 입구
 - 포트마다 다른 프로그램을 실행할 수 있다.
 - 3000번 포트로 index.js 프로그램을 실행할 수 있다.
- `app.get('/', ()=>{})`

사용자가 app 의 기본 폴더 (/) 로 들어왔을 때 function (req, res) {} 함수 내용을 실행 하겠다! 라는 뜻

사용자가 주소창에 localhost:3000 을 입력한 건 get 방식으로 '/' 디렉토리에 들어간 것을 의미함

그래서 콜백 함수를 실행하게 됨

콜백함수는 응답으로 Hello World 라는 글자를 출력함

 - get : HTTP 메소드
 - / : 라우팅
 - domain/about
 - domain/board 같은 것
 - () => {} : 콜백 함수
 - req : 요청 정보가 다 담겨 있음
 - res : 요청에 대한 응답 정보가 다 담겨 있음
 - 응답에 보내겠다. hello world 라는 문자를 담아서!
- 콜백 함수?
 - 기본 : 함수(파라미터)
 - 콜백 : 함수(끝나고 실행할 함수)
 - ex) setTimeout 함수
 - 보통 순서를 지정해 줄 때 사용
 - 나중에 실행 될 것을 함수에 넣어준다!

```
JS index.js x {} package.json
JS index.js > ...
1   setTimeout(()=>{console.log("5초 지남")}, 5000)

문제 출력 디버그 콘솔 터미널
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
^C
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
5초 지남
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
5초 지남
jeongseyeon@jeongseyeon-ui-MacBookPro node %
```

예제

GET /dog ⇒ { 'sound' : '멍멍' }

GET /cat ⇒ { 'sound' : '야옹' }

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

app.get('/', function (req, res) {
  res.send('Hello World')
})
app.get('/dog', function (req, res) {
  res.json({'sound':'멍멍'})
})
app.get('/cat', function (req, res) {
  res.json({'sound':'야옹'})
})

app.listen(3000)
```

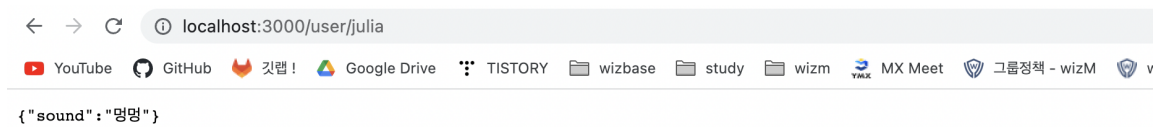
GET : params, query

- 변수 받기 (params)

```
6 }
7 app.get('/user/:id', function (req, res) {
8   const q = req.params
9   console.log(q)
10  res.json({'sound': '멍멍'})
11 })
12 app.get('/cat', function (req, res) {
13   res.json({'sound': '야옹'})
14 })
15
16 --- 14:44:33 (3000)

문제 출력 디버그 콘솔 터미널

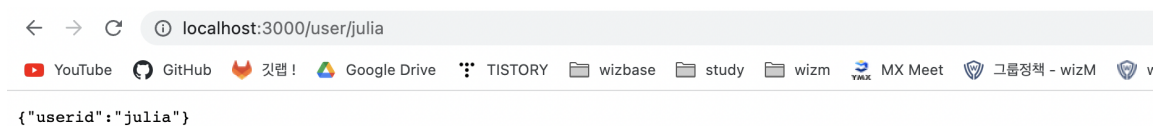
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
^C
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
5초 지남
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
5초 지남
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
^[[A^C
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
c^C
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
^C
jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
{ id: 'julic' }
{ id: 'julia' }
```



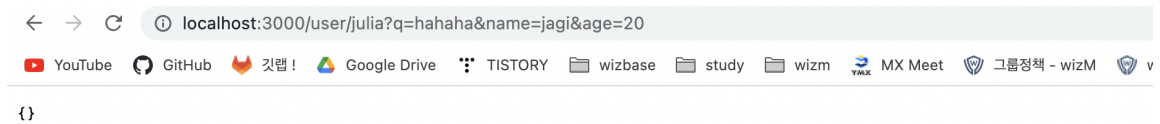
```
JS index.js > app.get('/user/:id') callback
1  const express = require('express')
2  const app = express()
3
4  app.get('/', function (req, res) {
5    res.send('Hello World')
6  })
7  app.get('/user/:id', function (req, res) {
8    const q = req.params
9    console.log(q)
10   res.json({'userid': q.id})
11 })
12 app.get('/cat', function (req, res) {
13   res.json({'sound': '야옹'})
14 })
15
16 --- 14:44:33 (3000)

문제 출력 디버그 콘솔 터미널

jeongseyeon@jeongseyeon-ui-MacBookPro node % node index.js
{ id: 'julia' }
```



- 변수 받기 (query)



동물 소리 API 서버 만들기

GET /sound/:name

name 따라서 다른 울음소리



- { 'name' : 'dog' } 로 들어올건데,
- { 종괄호 } 안에 KEY 값을 입력하면 더 간단해짐

```

14  app.get('/sound/:name', function (req, res) {
15      const { name } = req.params
16      if (name === "dog") {
17          res.json({'sound': '멍멍'})
18      } else if (name === "cat") {
19          res.json({'sound': '야옹'})
20      } else if (name === 'pig') {
21          res.json({'sound': '꿀꿀'})
22      } else {
23          res.json({'sound': '알수없음'})
24      }
25  })

```

CORS 이슈?

- html 파일로 서버에 요청 했을 때, html에 CORS 가 없으면 기본적으로 보안상 막음
- npm install cors
- 프론트에서 요청했을 때 오류가 없게 함
- app.use(cors()) 추가하면 완료 (line 5)

JS index.js > ...

```
1  const express = require('express')
2  var cors = require('cors')
3  const app = express()
4
5  app.use(cors())
6
7  app.get('/', function (req, res) {
8    res.send('Hello World')
9  })
10 app.get('/user/:id', function (req, res) {
11   // const q = req.params
12   // console.log(q)
13   const q = req.query
14   console.log(q)
15   res.json({'userid':q.name})
16 })
17 app.get(['/sound/:name', function (req, res) {
18   const { name } = req.params
19   if (name === "dog") {
20     res.json({'sound':'멍멍'})
21   } else if (name === "cat") {
22     res.json({'sound':'야옹'})
23   } else if (name === 'pig') {
24     res.json({'sound':'꿀꿀'})
25   } else {
26     res.json({'sound':'알수없음'})
27   }
28 })
29
```