Arduino-IOT

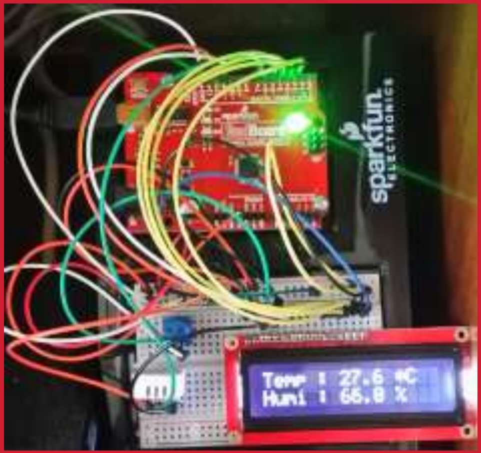
[wk03]

node.js express

Visualization of Signals using Arduino,

Node.js & storing signals in MongoDB

& mining data using Python



Drone-IoT-Comsi, INJE University

2nd semester, 2021

Email : chaos21c@gmail.com

My ID

ID를 확인하고 github에 repo 만들기

AA01	김준수	AA13	조재윤
AA02	김현서	AA14	고태승
AA03	박영훈	AA15	이한글
AA04	박윤호	AA16	장세진
AA05	성은지	AA17	장태호
AA06	손윤우	AA18	정지원
AA07	오세윤	AA19	진우태
AA08	우승철	AA20	황혁준
AA09	윤현석	AA21	장이제
AA10	이예주	AA22	박상현
AA11	강지환	AA23	정은성
AA12	성인제	AA24	김경영

위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 – AAnn

Public, README.md check

[Practice]

- [wk02]
  - Node module : aanninfo.js
  - Upload folder: aann-rpt02

wk02 : Practice-01 : AAnn_Rpt02

◆ **[Target of this week]**

My Info using node module – aanninfo.js

Upload folder : aann-rpt02

- 제출할 파일들

① **AAnn_package.png**

② **AAnn_HTTP.png**

③ **AAnn_TCP_Log.png**

④ **AAnn_Upload.png**

⑤ **AAnn_info.png**

⑥ start folder

⑦ **server folder**

Email : chaos21c@gmail.com , 제목: ID, 이름

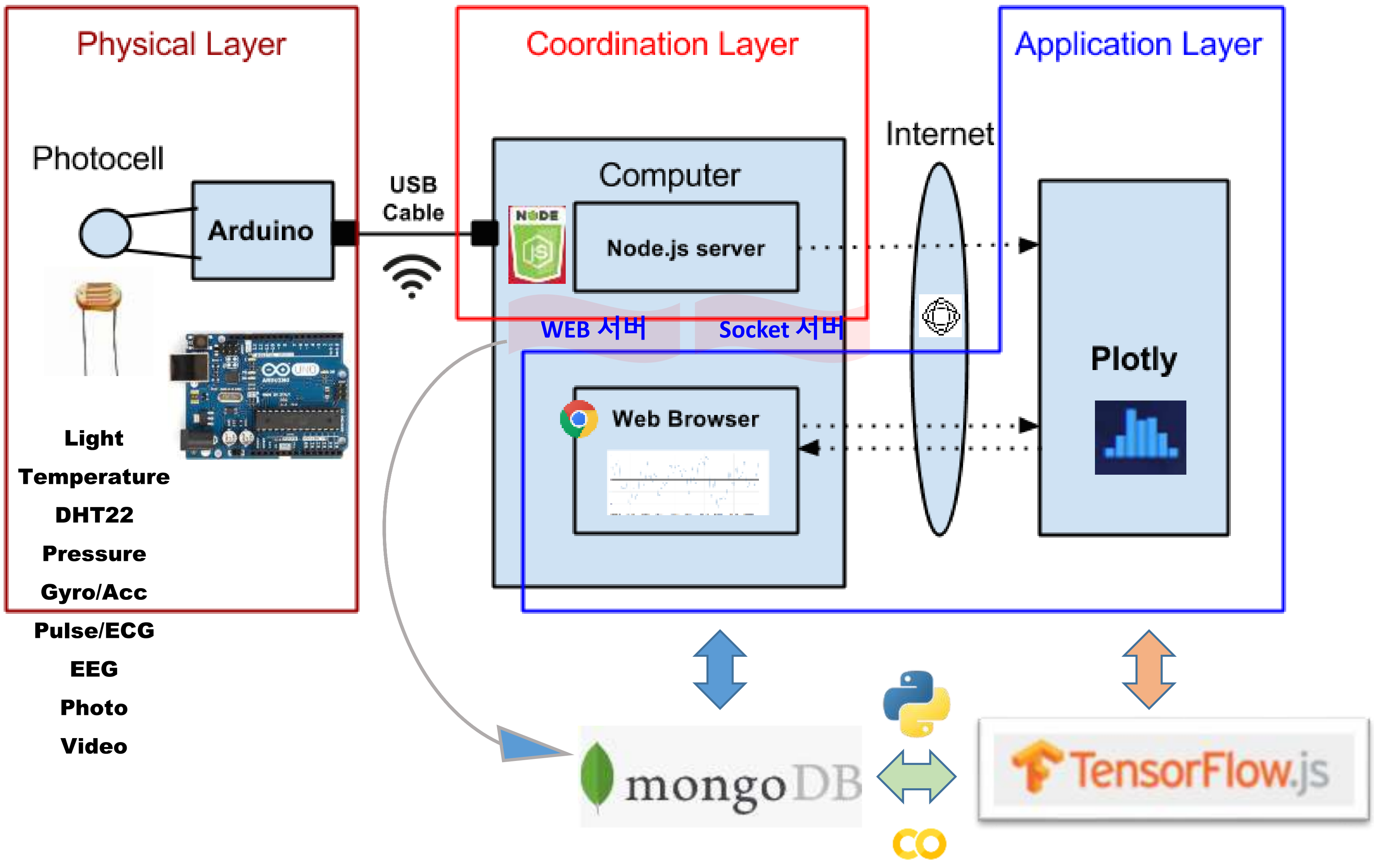
Purpose of AA

주요 수업 목표는 다음과 같다.

1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리

4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

Layout [H S C]



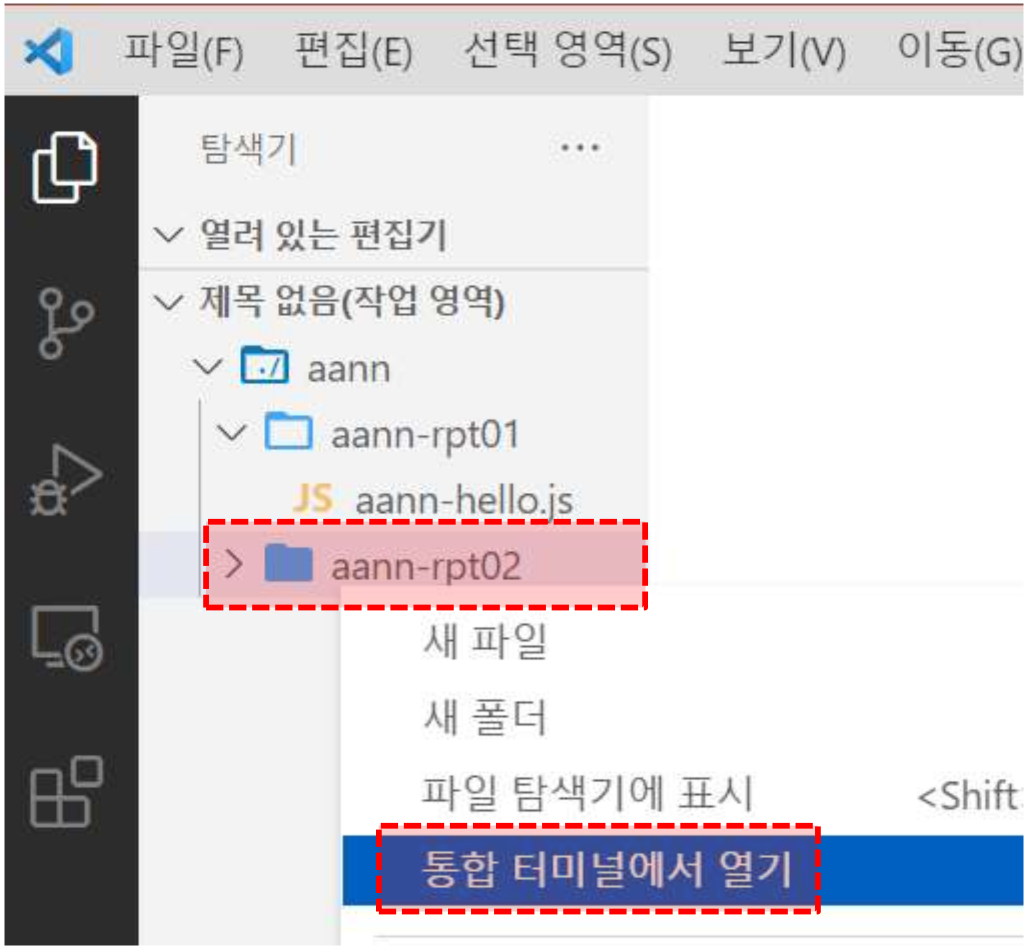
Physical Layer
Coordination Layer
Application Layer
Photocell
Arduino
USB
Cable
Computer
Node.js server
Internet
WEB 서버
Socket 서버
Web Browser
Plotly
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
mongoDB
TensorFlow.js

Node.js Project

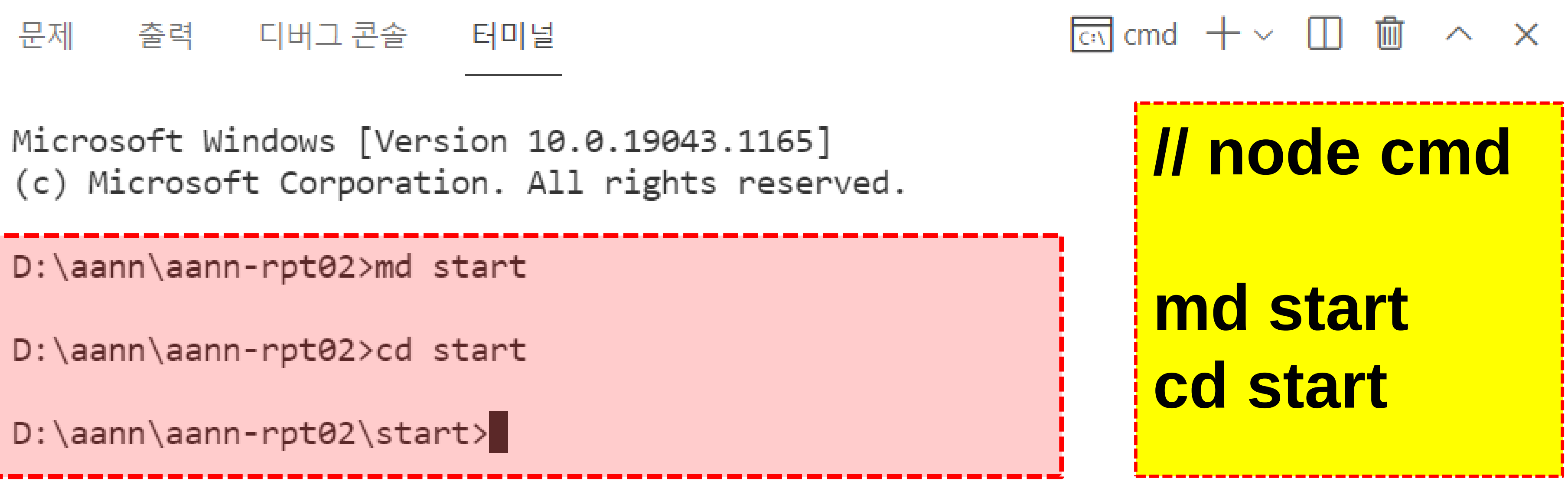
npm init

4.1 Node project : start terminal

// VScode
작업영역에
aann-rpt02
폴더 만들고
터미널 열기



4.1 Node project : start terminal

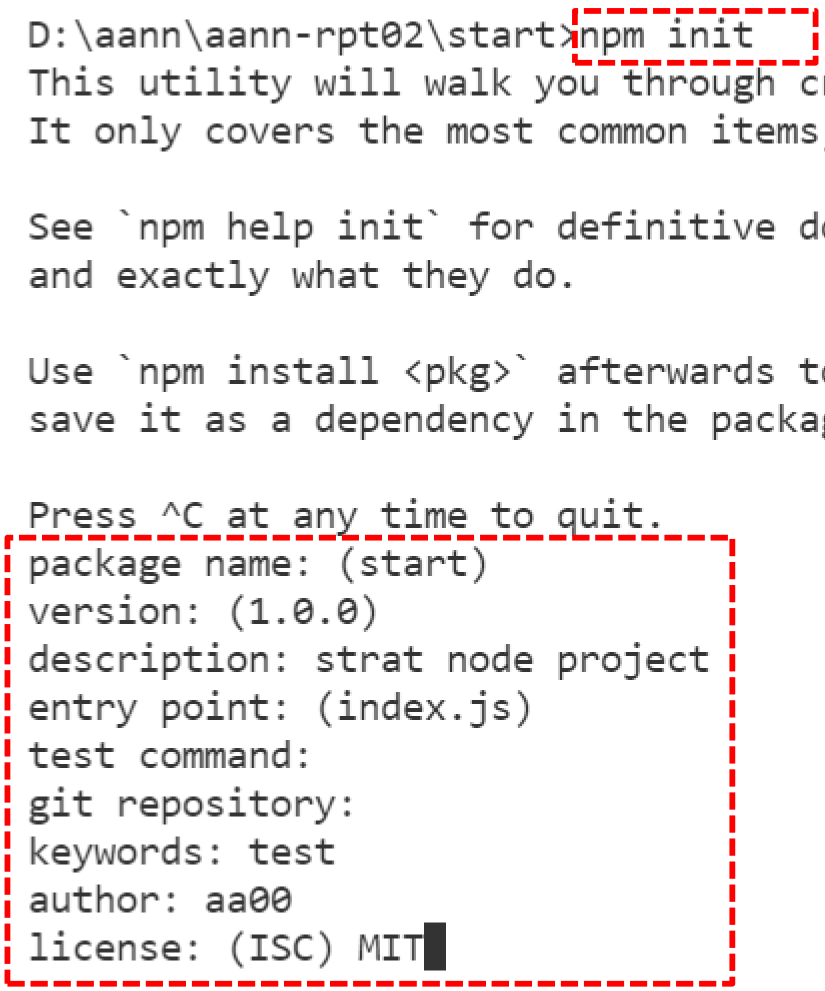


```
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

D:\aann\aann-rpt02>md start

D:\aann\aann-rpt02>cd start

D:\aann\aann-rpt02\start>
```

// node cmd

md start
cd start

4.2 Node project : npm init

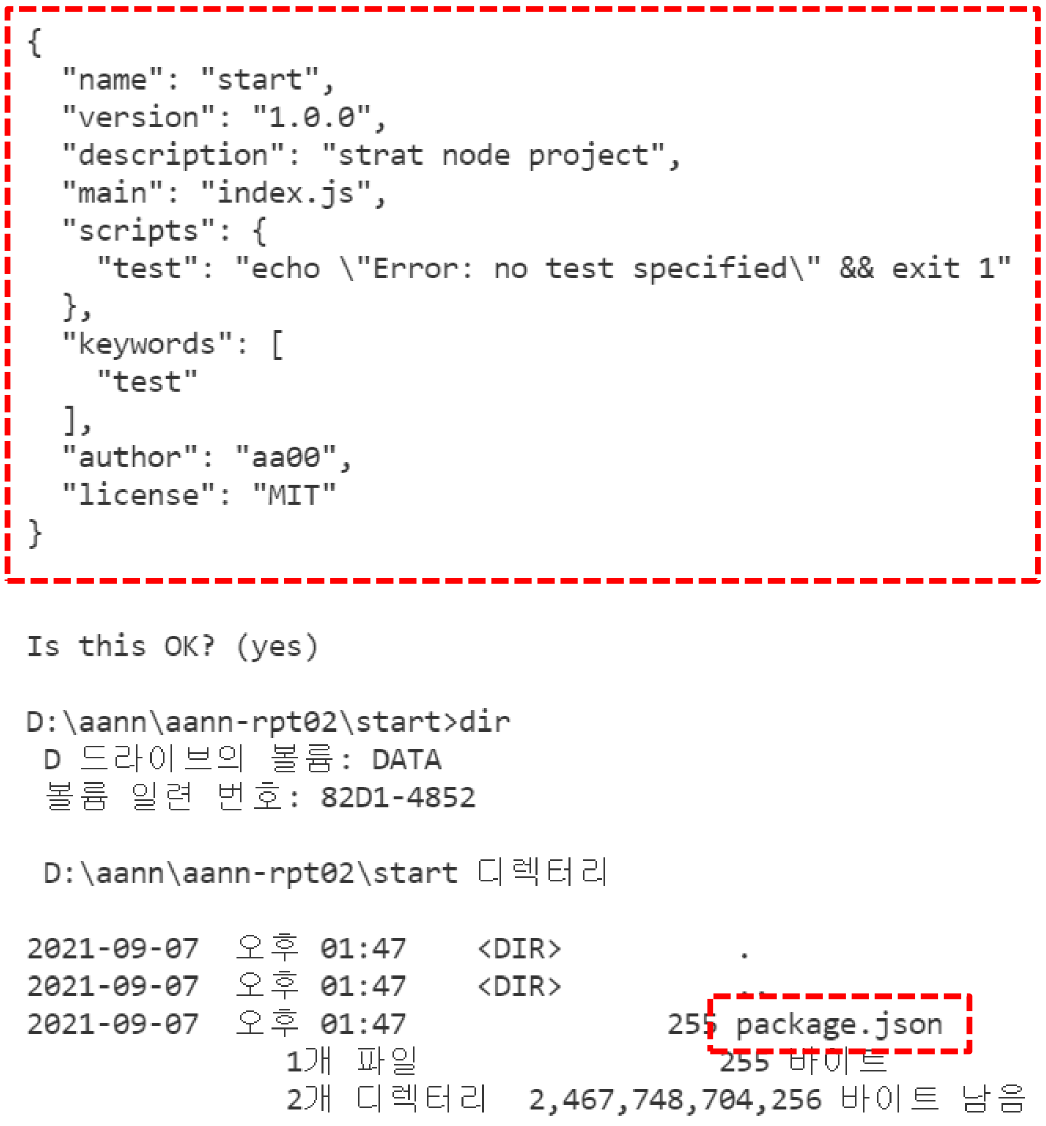


```
D:\aann\aann-rpt02\start>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

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: (start)
version: (1.0.0)
description: strat node project
entry point: (index.js)
test command:
git repository:
keywords: test
author: aa00
license: (ISC) MIT
```

// node project

npm init

4.3.1 Node project : package.json

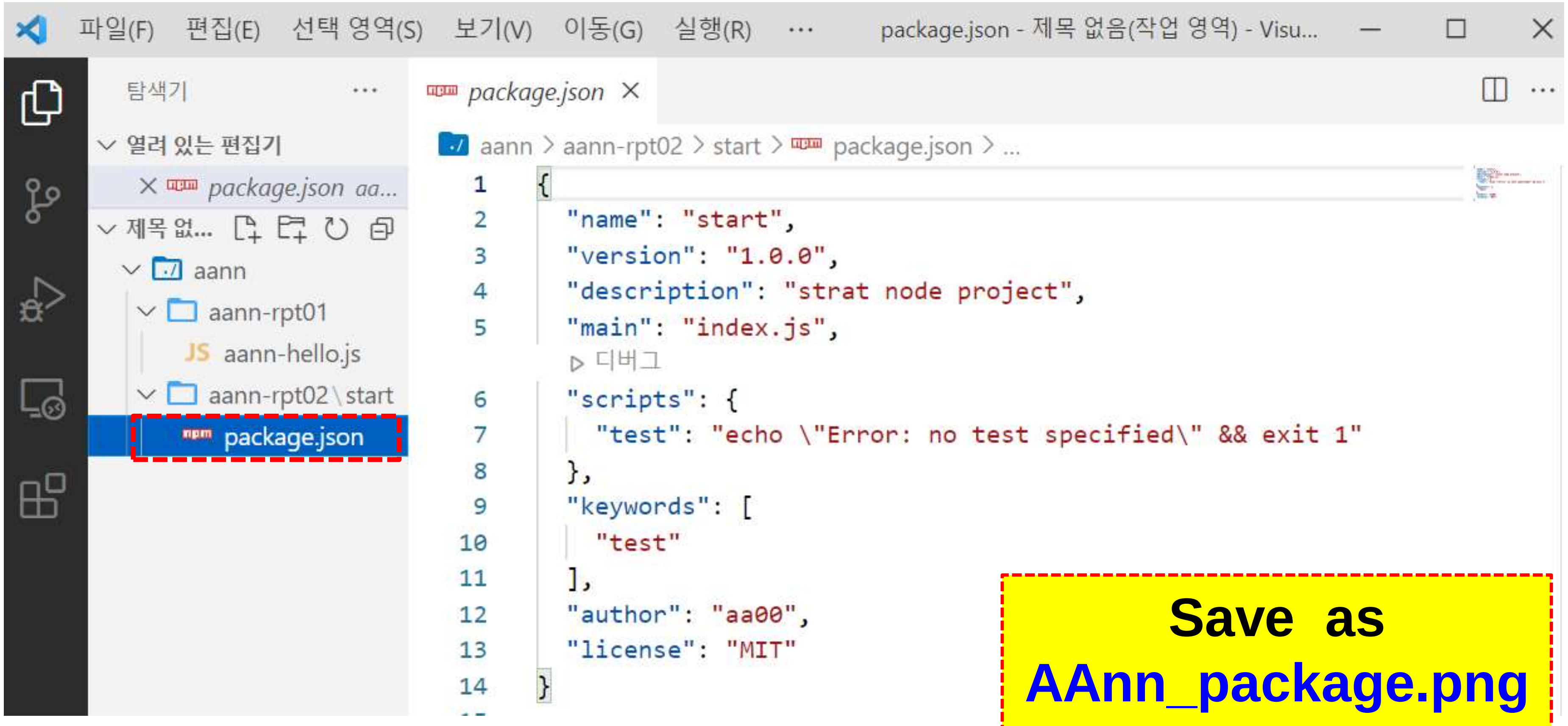


```
{
  "name": "start",
  "version": "1.0.0",
  "description": "strat node project",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "test"
  ],
  "author": "aa00",
  "license": "MIT"
}

Is this OK? (yes)

D:\aann\aann-rpt02\start>dir
 D 드라이브의 볼륨: DATA
 볼륨 일련 번호: 82D1-4852

 D:\aann\aann-rpt02\start 디렉터리

2021-09-07  오후 01:47    <DIR>          .
2021-09-07  오후 01:47    <DIR>          ..
2021-09-07  오후 01:47               255 package.json
               1개 파일                 255 바이트
               2개 디렉터리  2,467,748,704,256 바이트 남음
```

4.3.2 Node project : package.json

package.json

Node 프로젝트 설정 파일
- json file


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) ···
package.json - 제목 없음(작업 영역) - Visu...
탐색기
열려 있는 편집기
package.json aa...
제목 없...
aann
aann-rpt01
aann-hello.js
aann-rpt02 \ start
package.json
aann > aann-rpt02 > start > package.json > ...
▷ 디버그

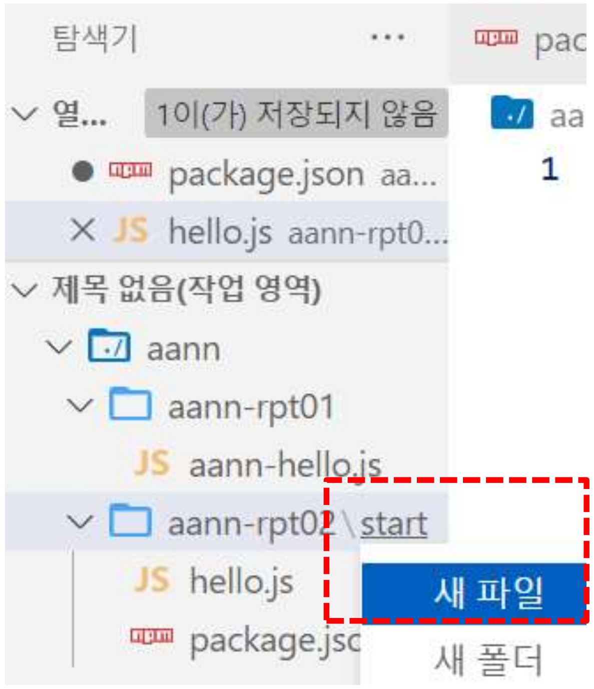
```json
{
  "name": "start",
  "version": "1.0.0",
  "description": "strat node project",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "test"
  ],
  "author": "aa00",
  "license": "MIT"
}
```

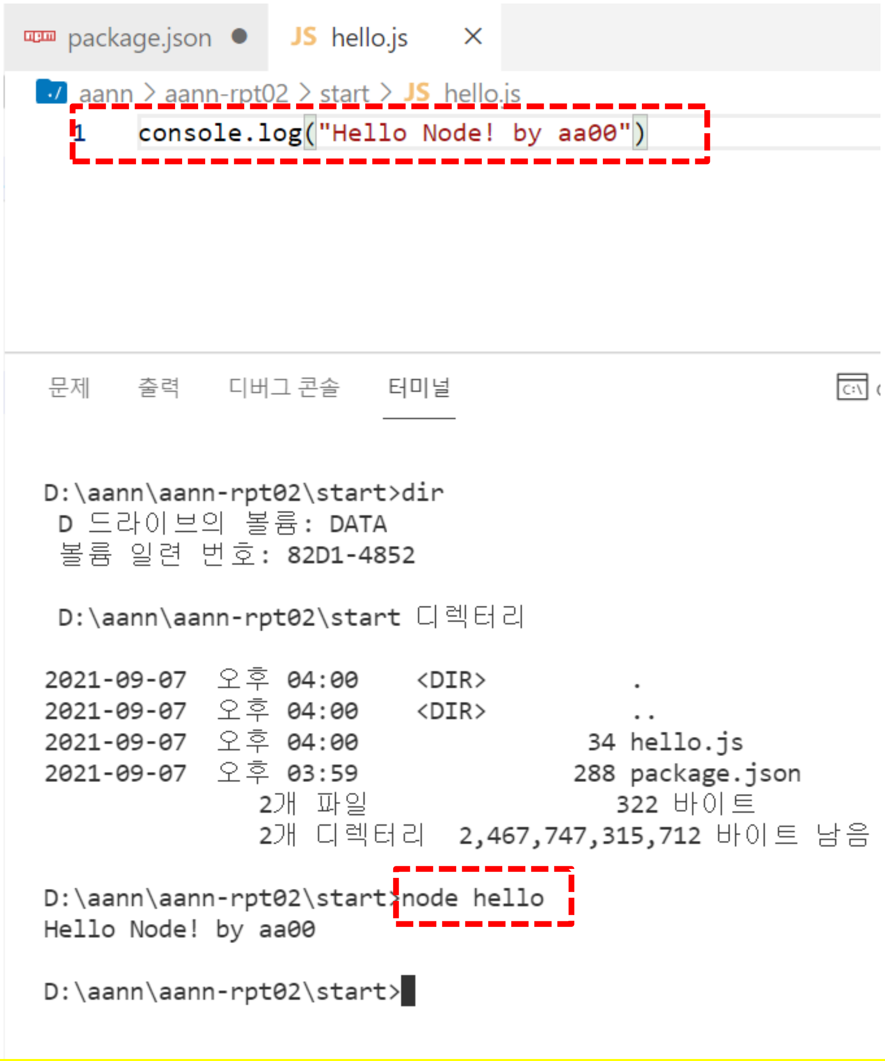
Save as
AAnn_package.png

5. Node Apps

Node Apps

5.1.1 Hello Node! – hello.js





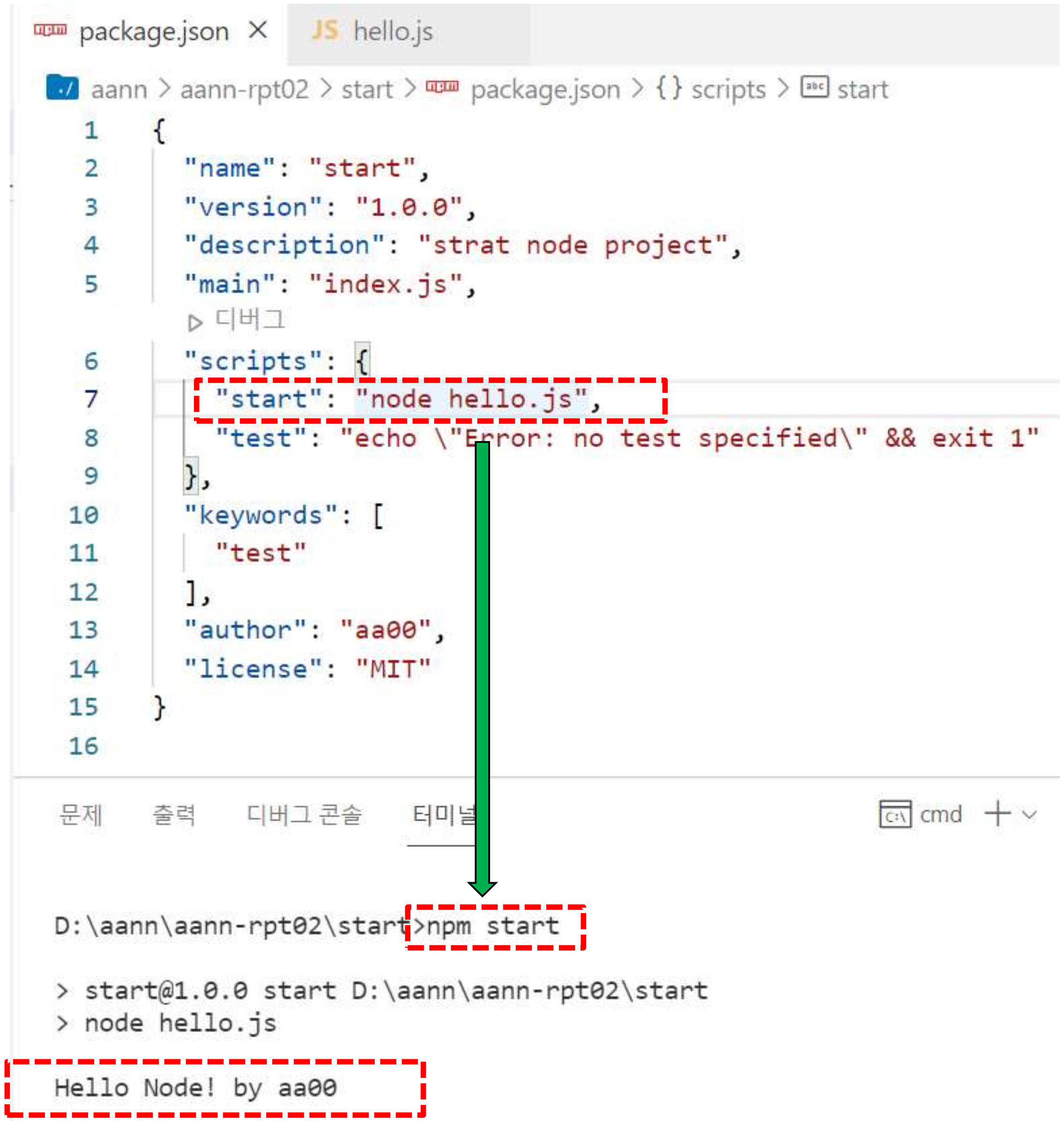
```
console.log("Hello Node! by aa00")
```

```
D:\aann\aann-rpt02\start>dir
 D 드라이브의 볼륨: DATA
 볼륨 일련 번호: 82D1-4852

 D:\aann\aann-rpt02\start 디렉터리

2021-09-07  오후 04:00    <DIR>          .
2021-09-07  오후 04:00    <DIR>          ..
2021-09-07  오후 04:00                34 hello.js
2021-09-07  오후 03:59               288 package.json
               2개 파일                 322 바이트
               2개 디렉터리  2,467,747,315,712 바이트 남음

D:\aann\aann-rpt02\start>node hello
Hello Node! by aa00

D:\aann\aann-rpt02\start>
```

start 폴더 안에 hello.js 파일

VSCode 터미널에서 실행 : node js-file-name

5.1.2 Hello Node! – npm start 로 실행



package.json
hello.js
aann > aann-rpt02 > start > package.json > {} scripts > start
▷ 디버그
문제
출력
디버그 콘솔
터미널
cmd

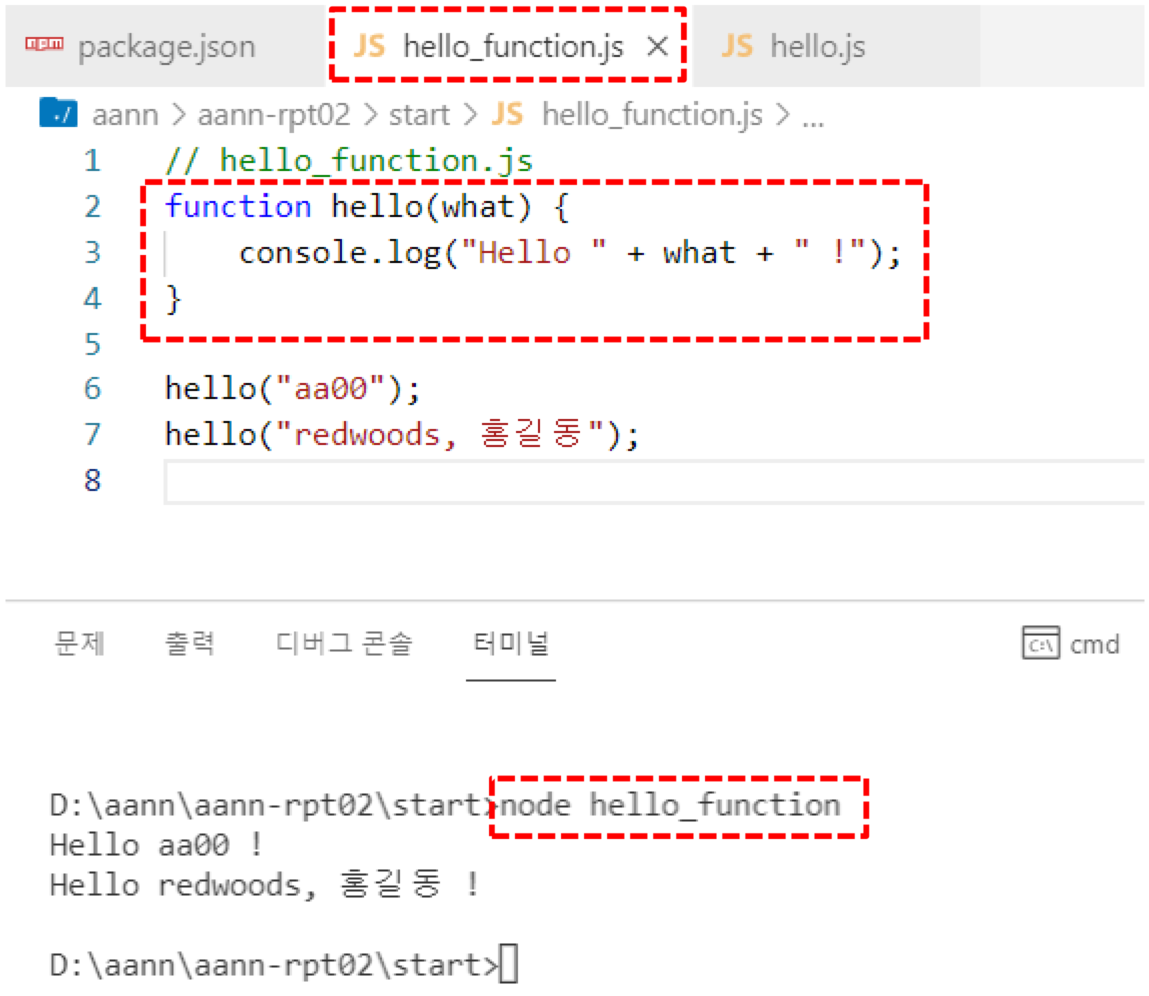
```json
{
  "name": "start",
  "version": "1.0.0",
  "description": "strat node project",
  "main": "index.js",
  "scripts": {
    "start": "node hello.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "test"
  ],
  "author": "aa00",
  "license": "MIT"
}
```

```
D:\aann\aann-rpt02\start>npm start

> start@1.0.0 start D:\aann\aann-rpt02\start
> node hello.js

Hello Node! by aa00
```

5.2 Node Apps – function

Using function in node

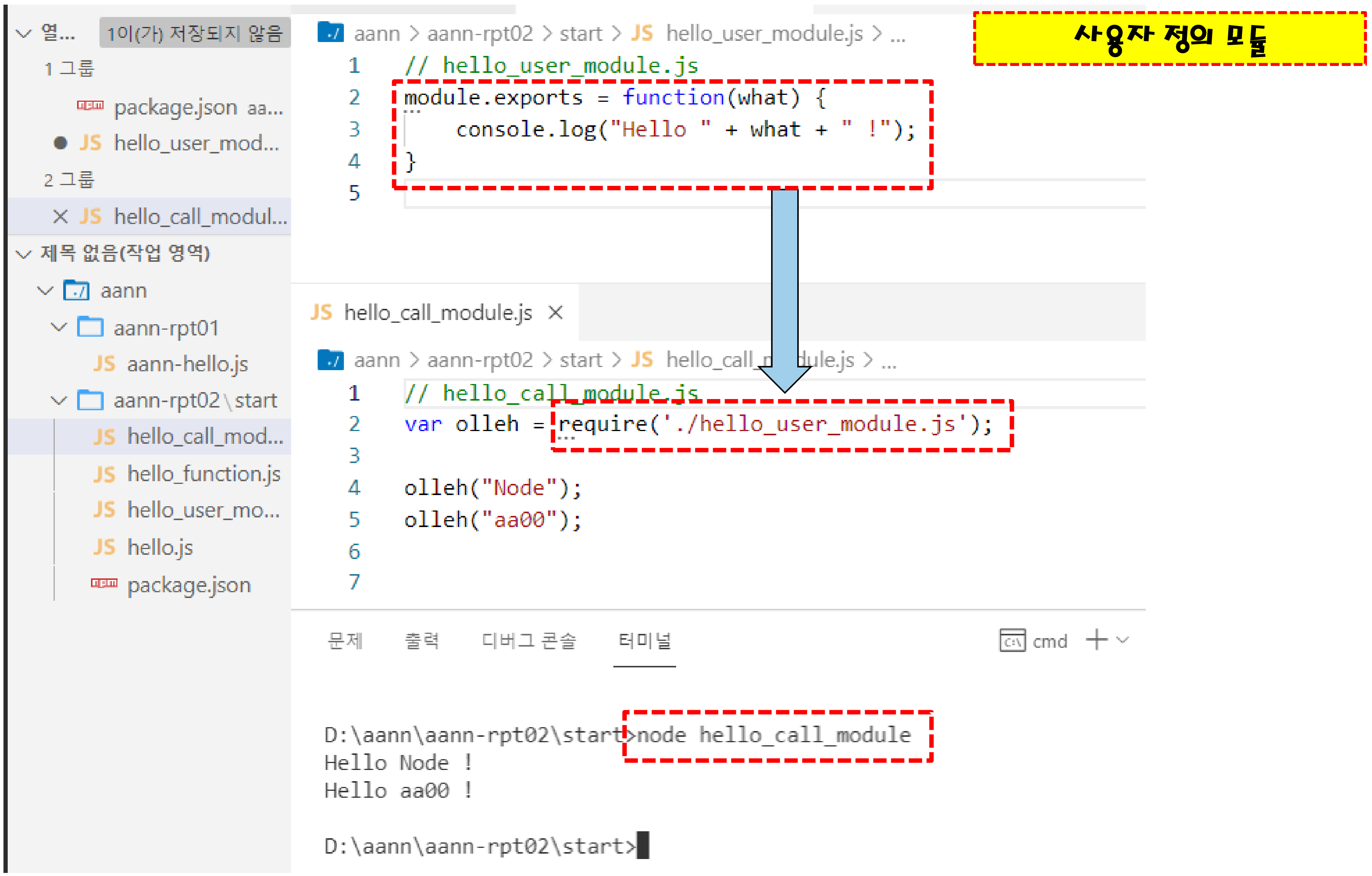
5.2.1 Using function in node



```
// hello_function.js
function hello(what) {
    console.log("Hello " + what + " !");
}

hello("aa00");
hello("redwoods, 홍길동");
```

```
D:\aann\aann-rpt02\start>node hello_function
Hello aa00 !
Hello redwoods, 홍길동 !

D:\aann\aann-rpt02\start>
```

5.2.2 Using user-module: hello_user_module.js



```javascript
// hello_user_module.js
module.exports = function(what) {
    console.log("Hello " + what + " !");
}
```

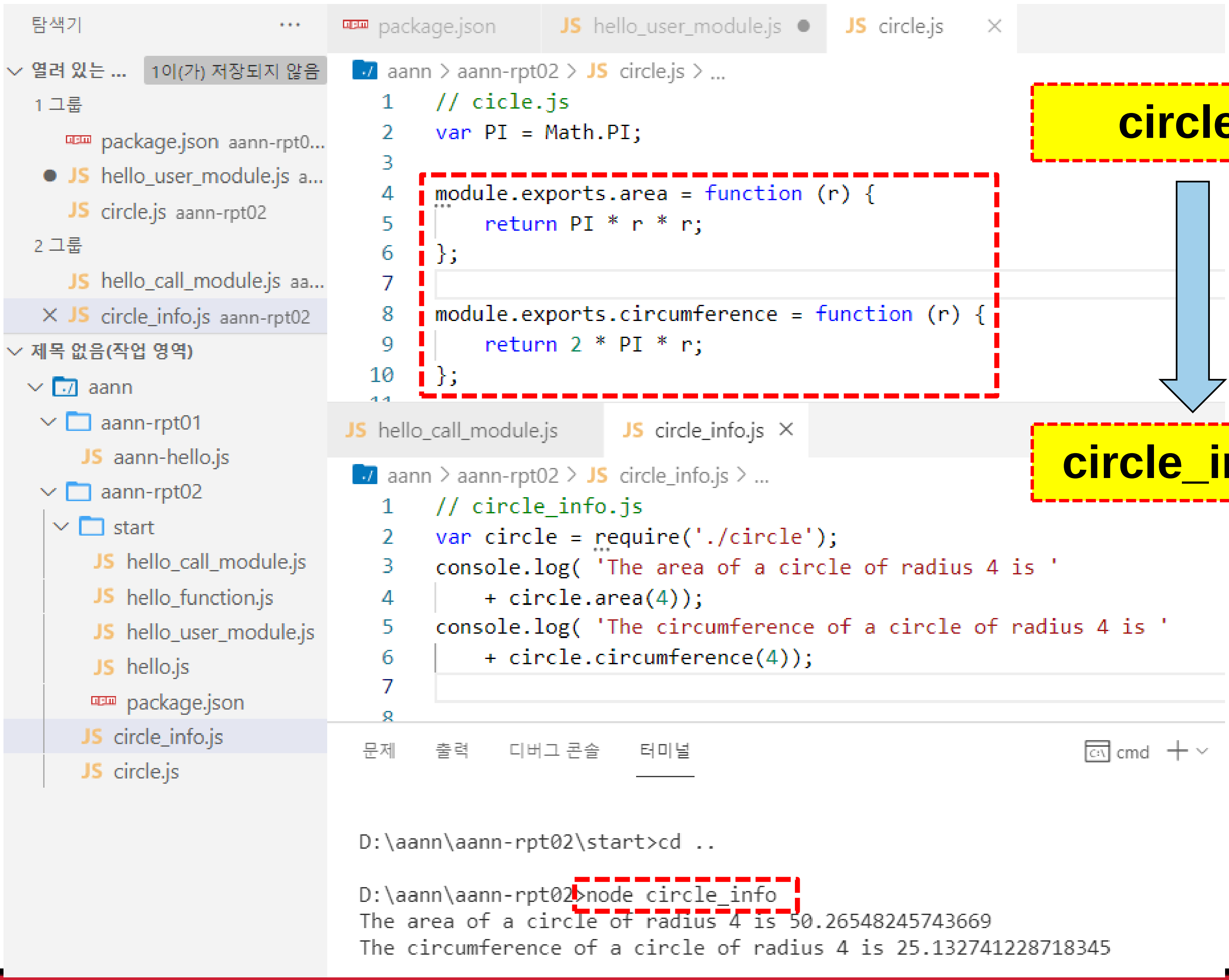
```javascript
// hello_call_module.js
var olleh = require('./hello_user_module.js');

olleh("Node");
olleh("aa00");
```

```
D:\aann\aann-rpt02\start>node hello_call_module
Hello Node !
Hello aa00 !

D:\aann\aann-rpt02\start>
```

Call user-module 'hello_user_module.js' from hello_call_module.js

[extra code] local module : circle.js

circle_info.js uses local module circle.js.



circle.js

```javascript
// cicle.js
var PI = Math.PI;

module.exports.area = function (r) {
    return PI * r * r;
};

module.exports.circumference = function (r) {
    return 2 * PI * r;
};
```

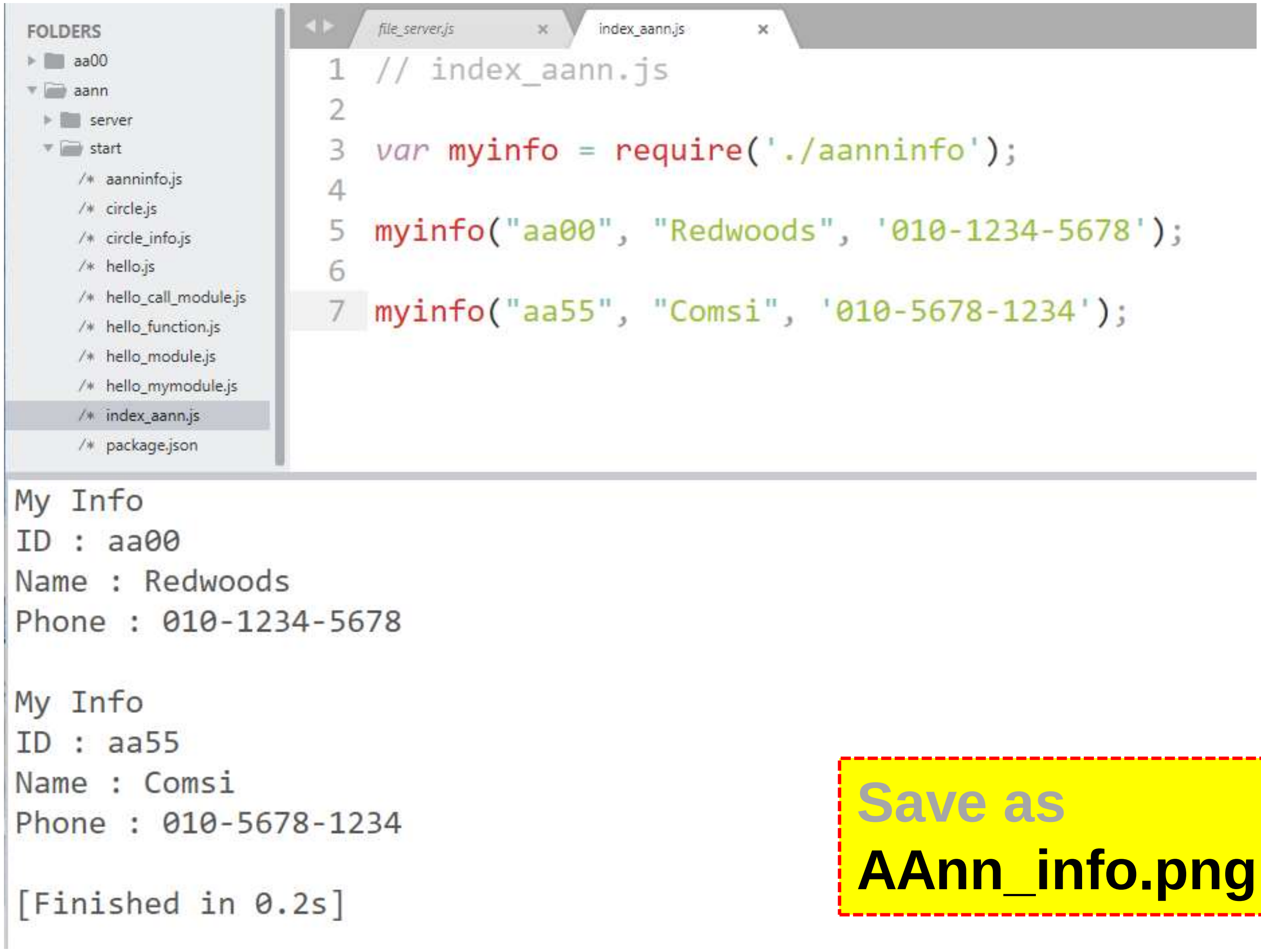
circle_info.js

```javascript
// circle_info.js
var circle = require('./circle');
console.log( 'The area of a circle of radius 4 is '
    + circle.area(4));
console.log( 'The circumference of a circle of radius 4 is '
    + circle.circumference(4));
```

```
D:\aann\aann-rpt02\start>cd ..

D:\aann\aann-rpt02>node circle_info
The area of a circle of radius 4 is 50.26548245743669
The circumference of a circle of radius 4 is 25.132741228718345
```

[practice] local module : aanninfo.js

index_aann.js uses local module aanninfo.js in start subfolder.



```
// index_aann.js

var myinfo = require('./aanninfo');

myinfo("aa00", "Redwoods", '010-1234-5678');

myinfo("aa55", "Comsi", '010-5678-1234');
```

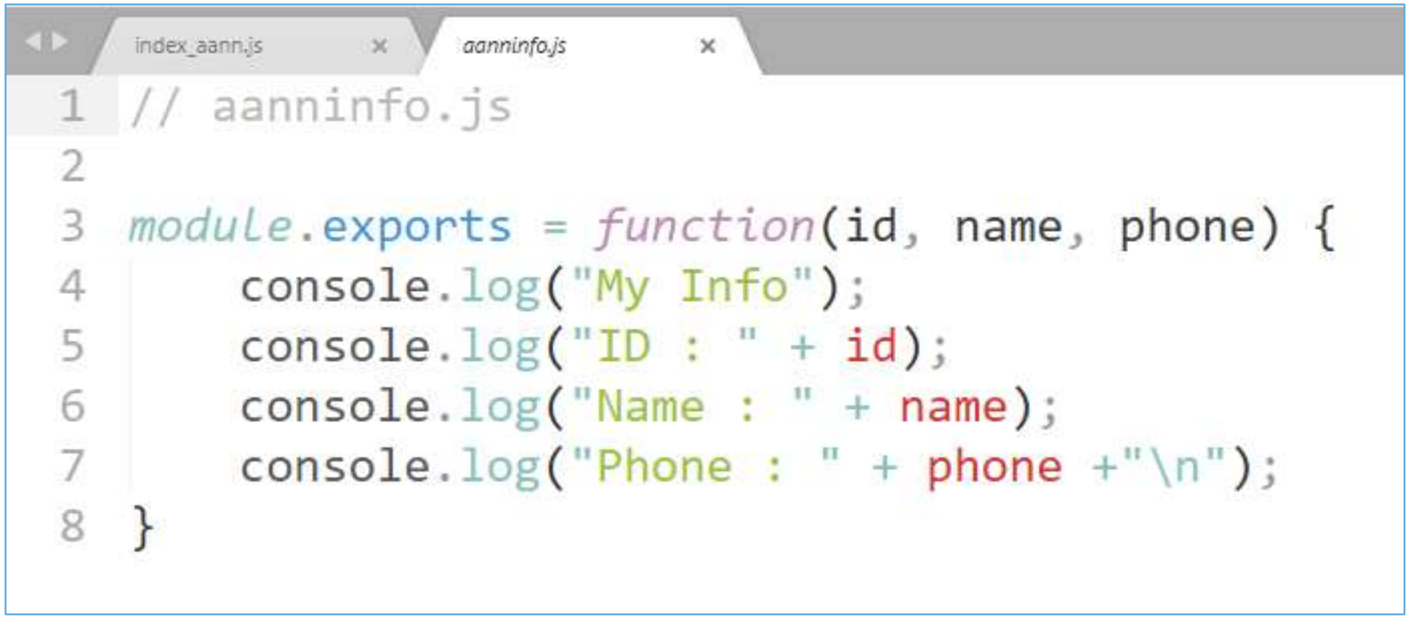
```
My Info
ID : aa00
Name : Redwoods
Phone : 010-1234-5678

My Info
ID : aa55
Name : Comsi
Phone : 010-5678-1234

[Finished in 0.2s]
```

Save as
AAnn_info.png

[practice] local module : aanninfo.js

How to make aanninfo.js in start subfolder.

1. **Make local module – aanninfo.js**
2. **Call aanninfo.js from index_aann.js.**
3. **Capture your result.**



```
// aanninfo.js

module.exports = function(id, name, phone) {
    console.log("My Info");
    console.log("ID : " + id);
    console.log("Name : " + name);
    console.log("Phone : " + phone +"\n");
}
```

[참고] Node local module 만들기

Node server

Node.js Server

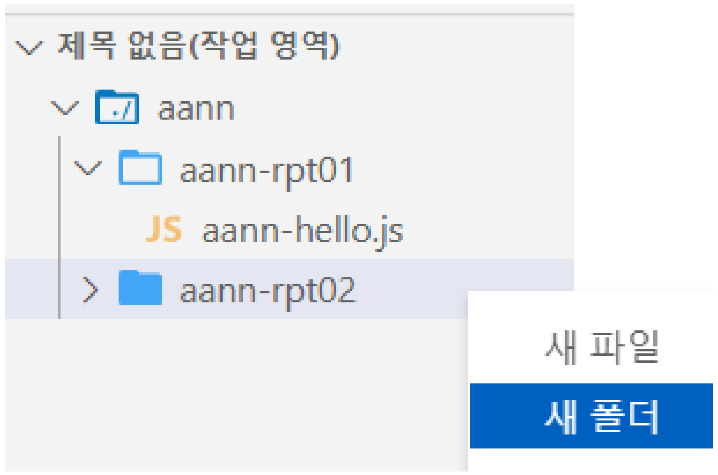
1. http, tcp, file
2. Express

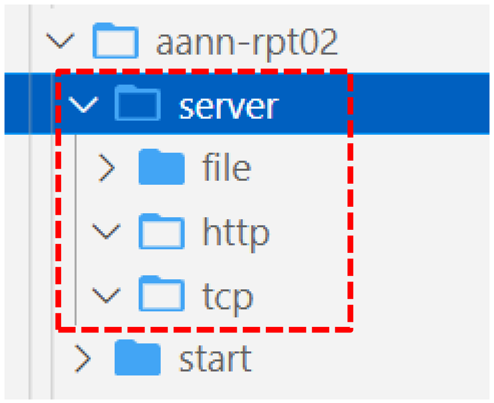
6. Node Server

Node Server I.

1. **HTTP server**
2. **TCP server**
3. **File upload**

6.1 Node server : working folders



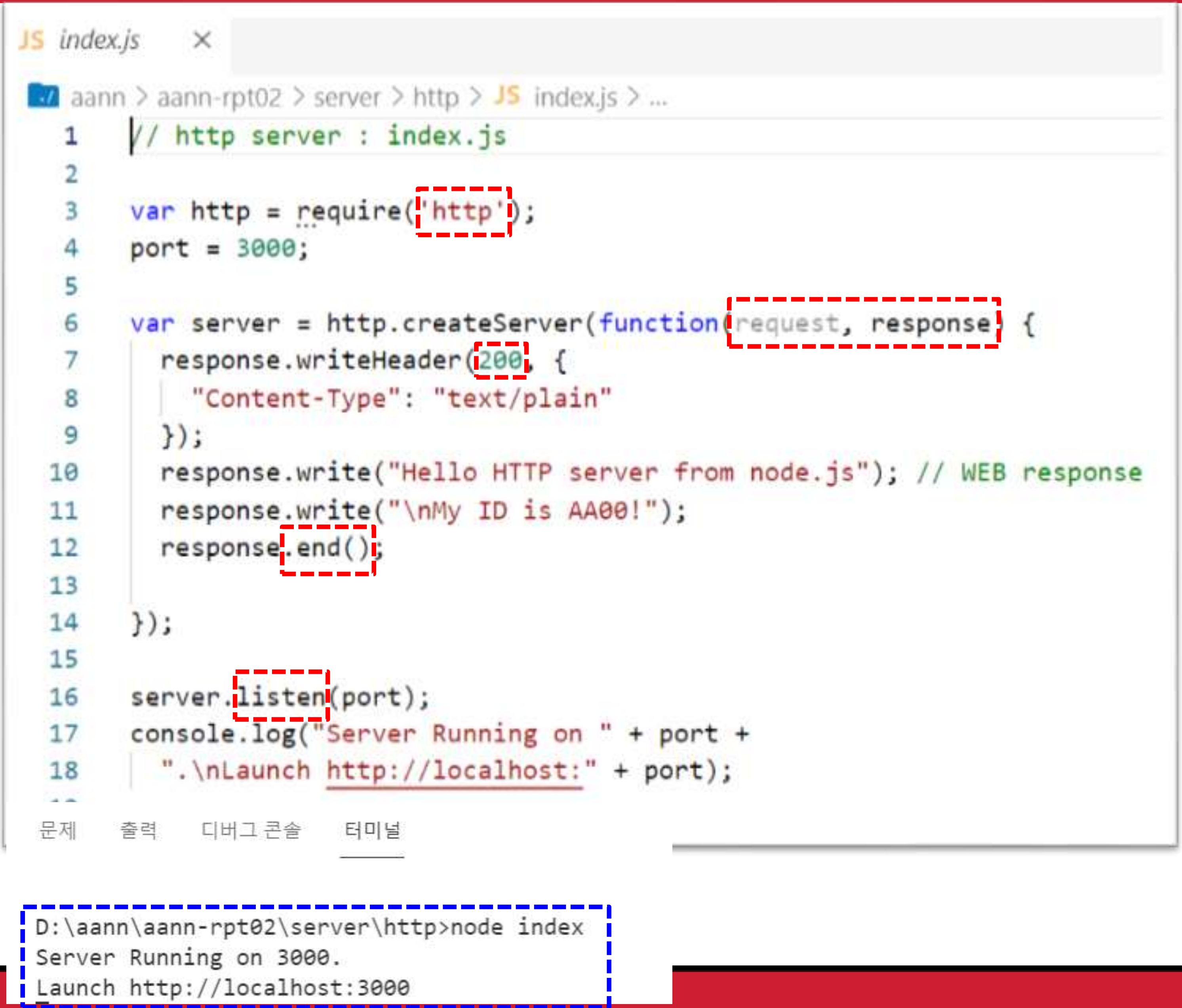


6. Node Server

Node Server I.

1. **HTTP server**
2. TCP server
3. File upload

6.1.1 http server



JS index.js ×

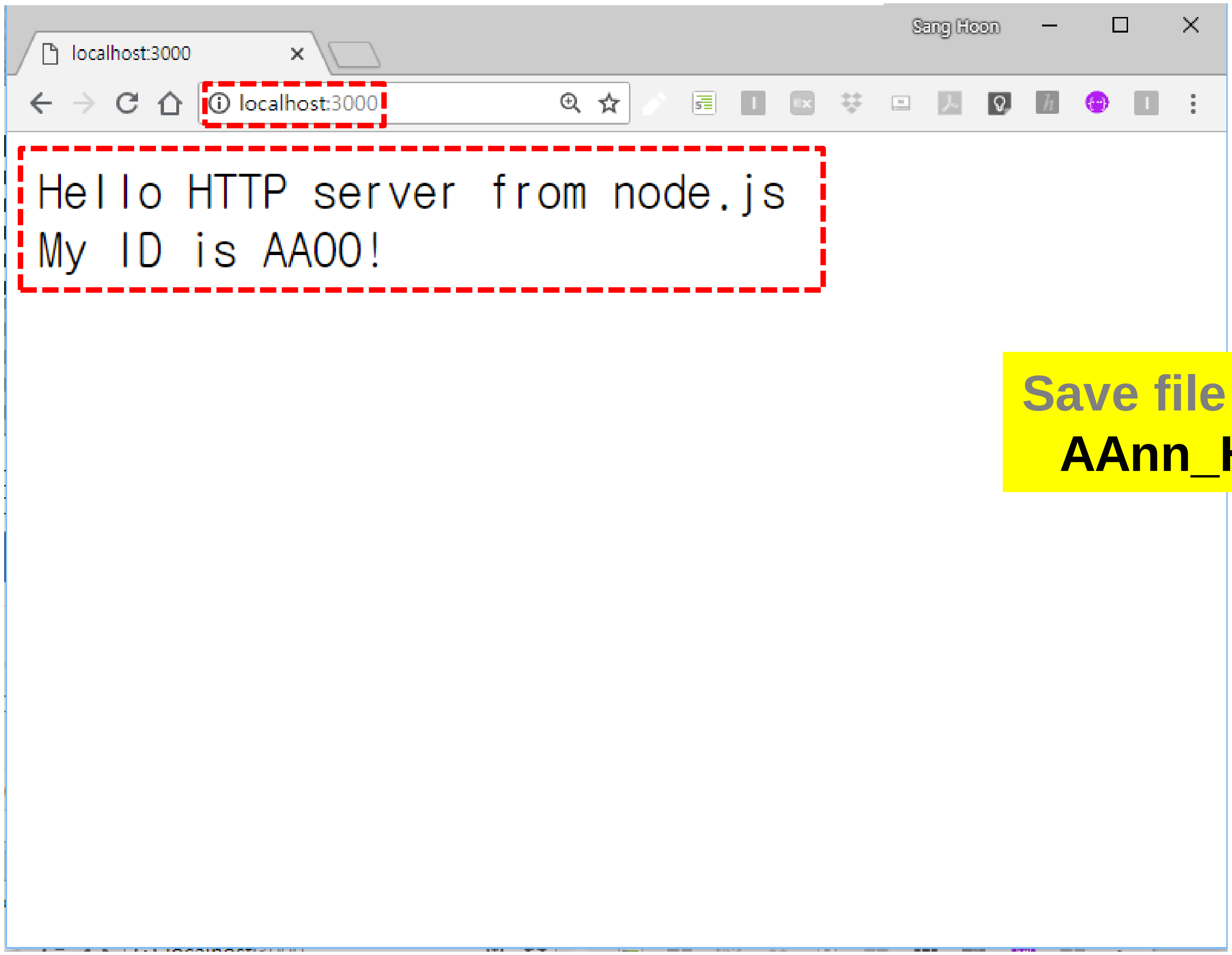
aann > aann-rpt02 > server > http > JS index.js > ...

```
// http server : index.js

var http = require('http');
port = 3000;

var server = http.createServer(function(request, response) {
  response.writeHeader(200, {
    "Content-Type": "text/plain"
  });
  response.write("Hello HTTP server from node.js"); // WEB response
  response.write("\nMy ID is AA00!");
  response.end();

});

server.listen(port);
console.log("Server Running on " + port +
  ".\nLaunch http://localhost:" + port);
```

```
D:\aann\aann-rpt02\server\http>node index
Server Running on 3000.
Launch http://localhost:3000
```

6.1.2 http server : result 1

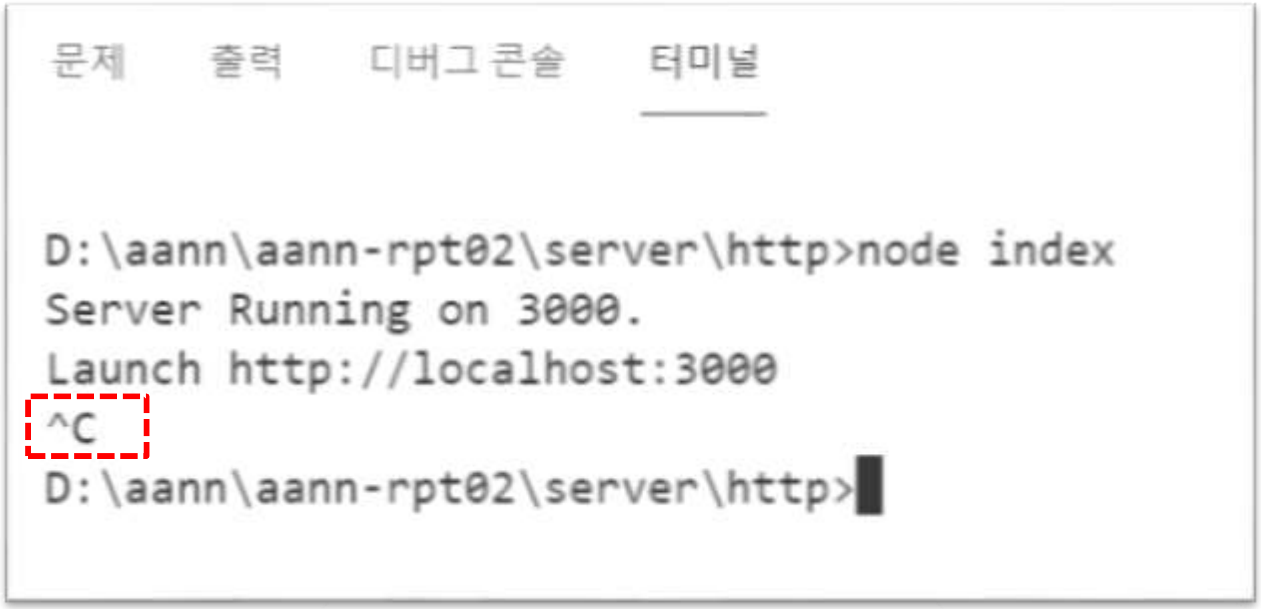
```
Server Running on 3000.
Launch http://localhost:3000
```



```
Hello HTTP server from node.js
My ID is AA00!
```

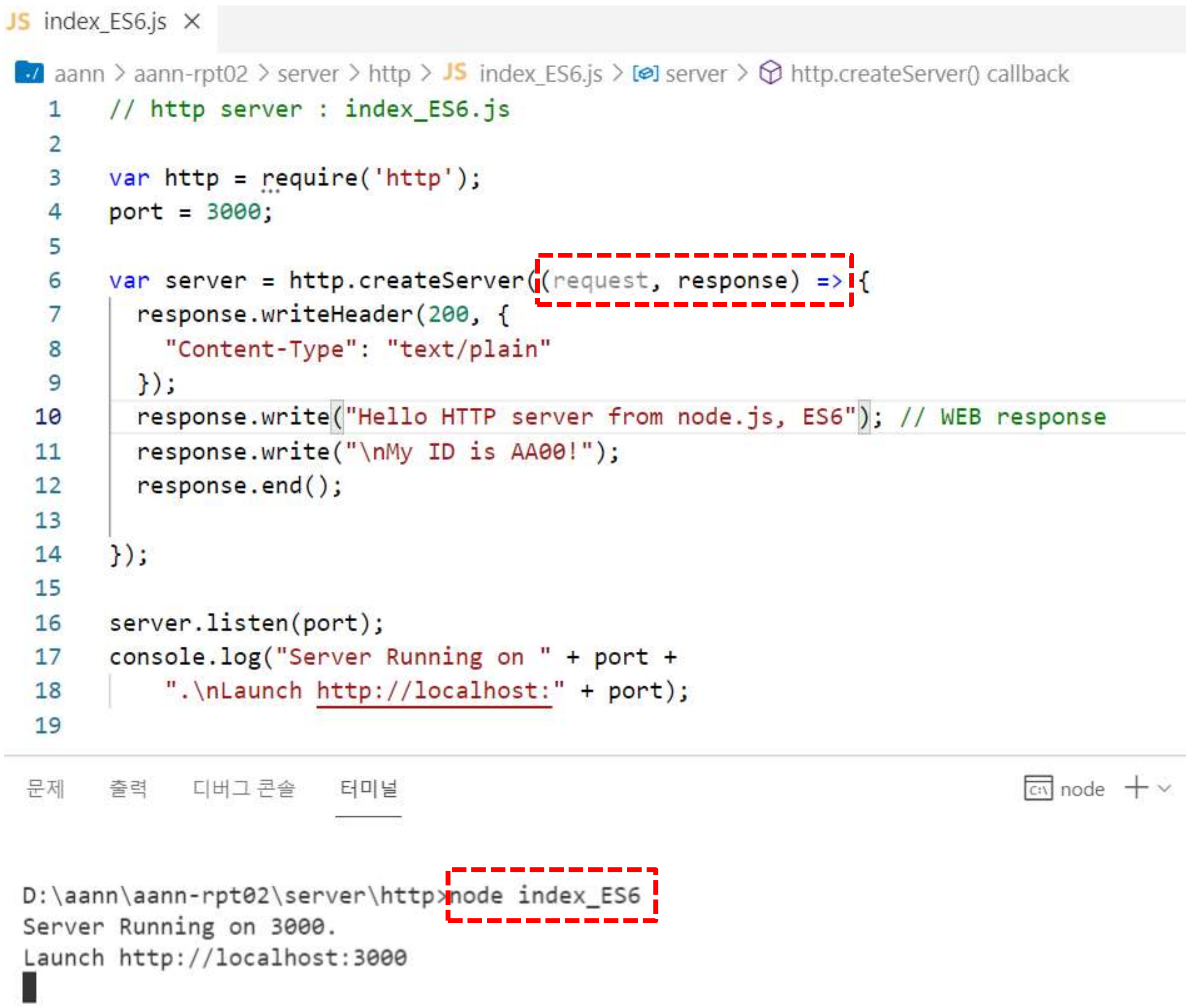
Save file
AAnn_HTTP.png

6.1.3 http server – stop server !!! → ^C



```
D:\aann\aann-rpt02\server\http>node index
Server Running on 3000.
Launch http://localhost:3000
^C
D:\aann\aann-rpt02\server\http>
```

6.1.4 http server – ES6 version


JS index_ES6.js ×
aann > aann-rpt02 > server > http > JS index_ES6.js > server > http.createServer() callback
문제
출력
디버그 콘솔
터미널
node

```javascript
// http server : index_ES6.js

var http = require('http');
port = 3000;

var server = http.createServer((request, response) =>{
  response.writeHeader(200, {
    "Content-Type": "text/plain"
  });
  response.write("Hello HTTP server from node.js, ES6"); // WEB response
  response.write("\nMy ID is AA00!");
  response.end();

});

server.listen(port);
console.log("Server Running on " + port +
    ".\nLaunch http://localhost:" + port);
```

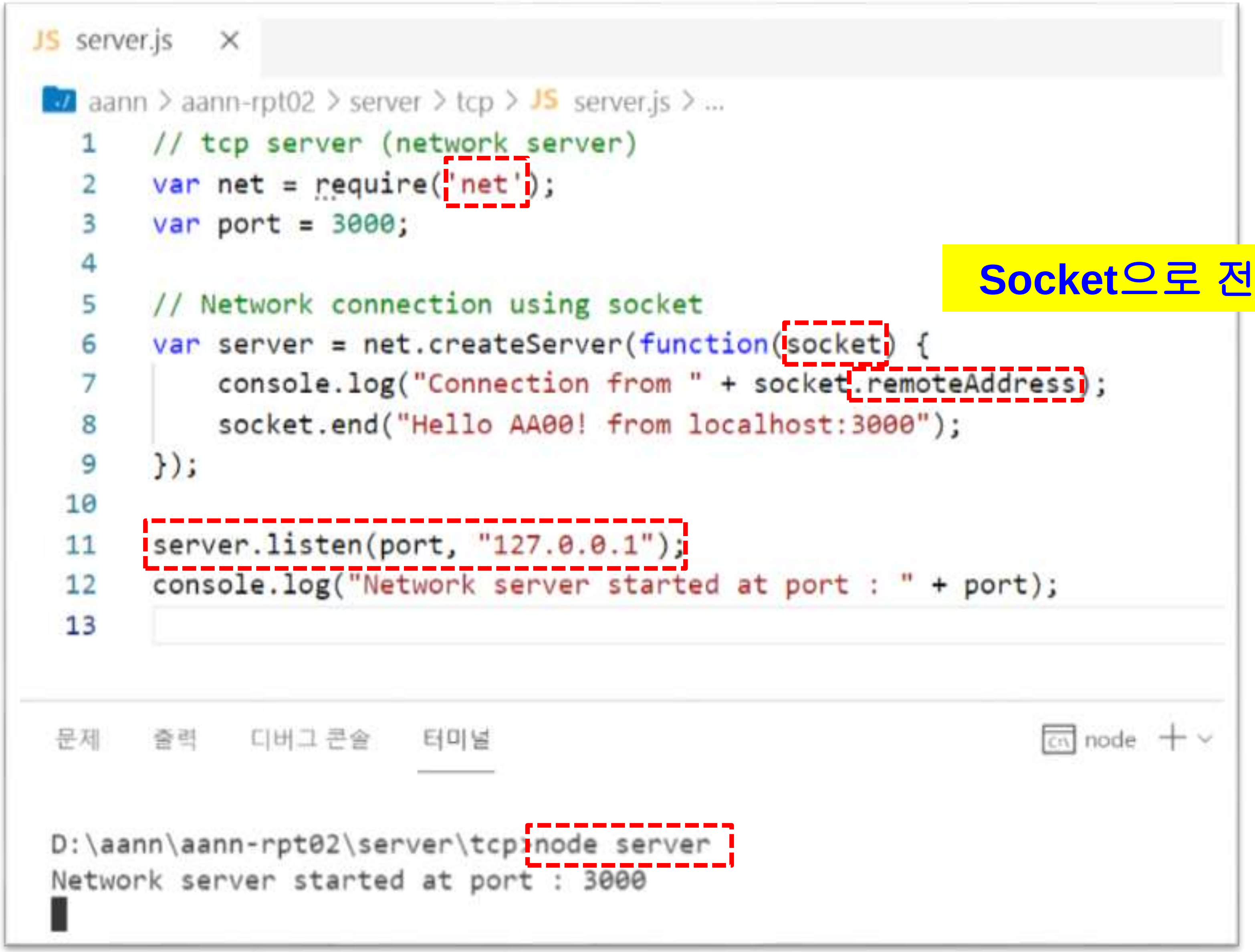
```
D:\aann\aann-rpt02\server\http>node index_ES6
Server Running on 3000.
Launch http://localhost:3000
```

6. Node Server

Node Server I.

1. HTTP server
2. **TCP server**
3. File upload

6.2.1 tcp server (socket connection)



server.js

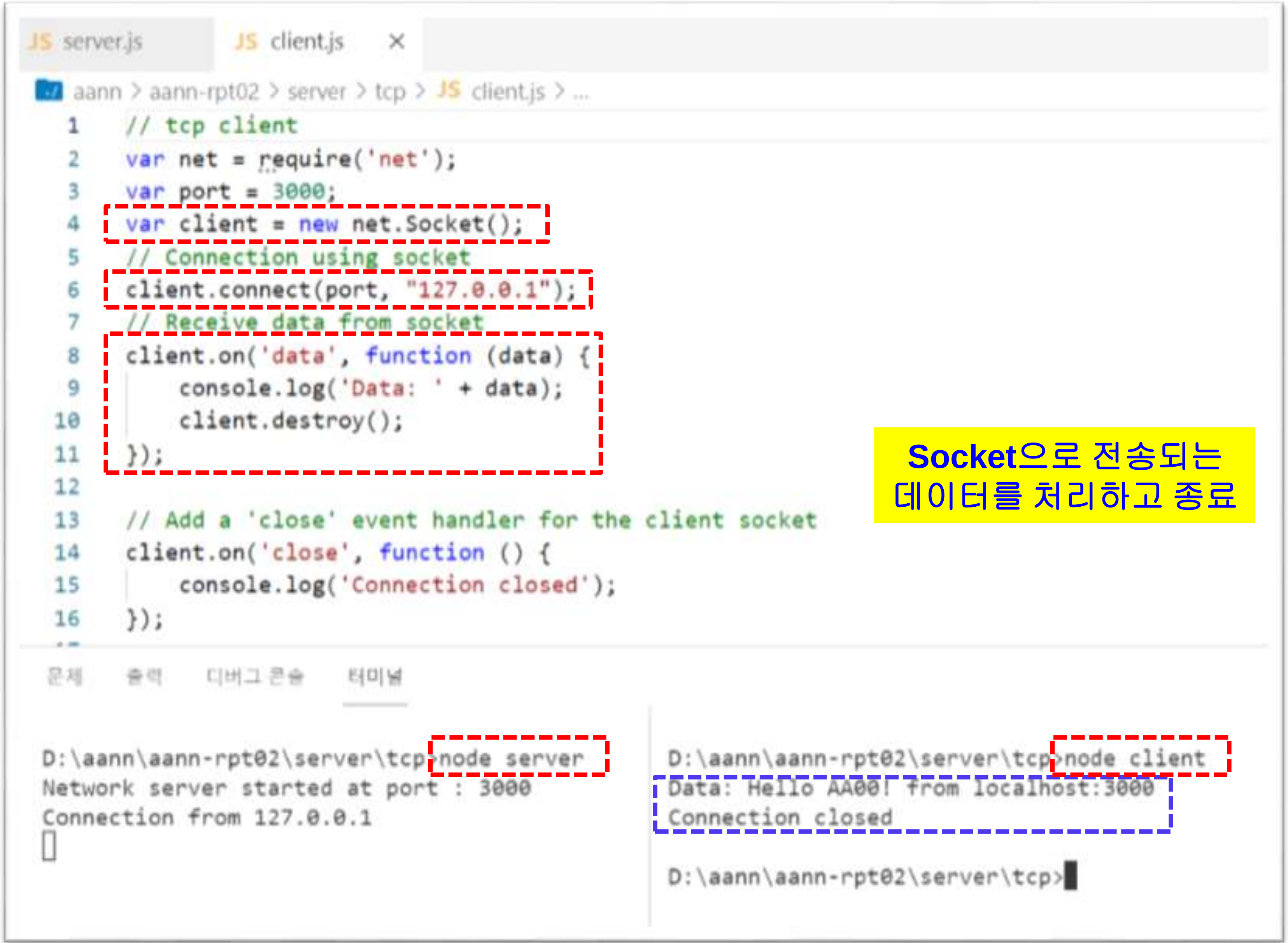
aann > aann-rpt02 > server > tcp > server.js > ...

```
// tcp server (network server)
var net = require('net');
var port = 3000;

// Network connection using socket
var server = net.createServer(function(socket) {
    console.log("Connection from " + socket.remoteAddress);
    socket.end("Hello AA00! from localhost:3000");
});

server.listen(port, "127.0.0.1");
console.log("Network server started at port : " + port);
```

```
D:\aann\aann-rpt02\server\tcp>node server
Network server started at port : 3000
█
```

Socket으로 전송

6.2.2 tcp client



server.js   client.js

aann > aann-rpt02 > server > tcp > client.js > ...

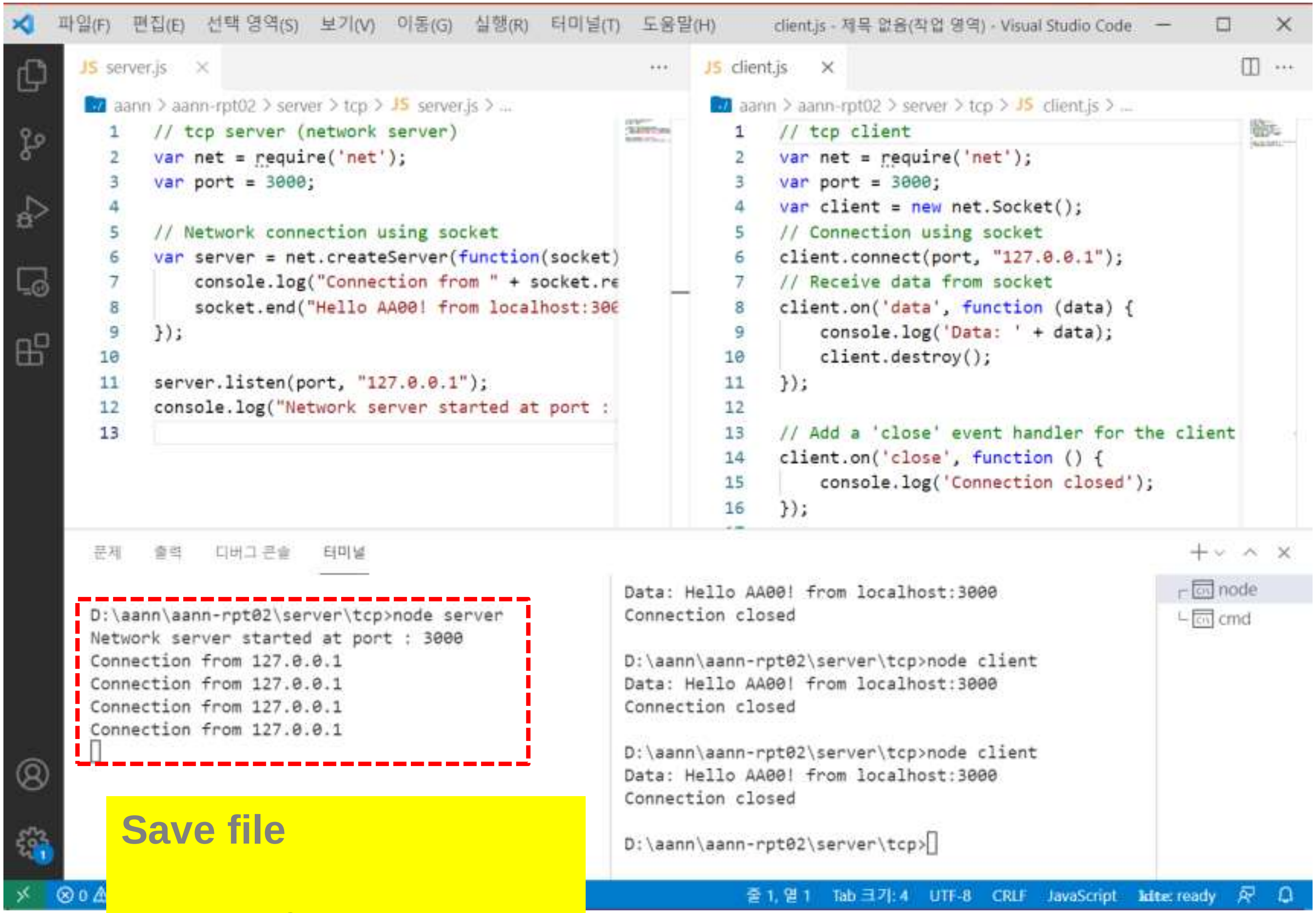
```
// tcp client
var net = require('net');
var port = 3000;
var client = new net.Socket();
// Connection using socket
client.connect(port, "127.0.0.1");
// Receive data from socket
client.on('data', function (data) {
    console.log('Data: ' + data);
    client.destroy();
});

// Add a 'close' event handler for the client socket
client.on('close', function () {
    console.log('Connection closed');
});
```

```
D:\aann\aann-rpt02\server\tcp>node server
Network server started at port : 3000
Connection from 127.0.0.1
```

```
D:\aann\aann-rpt02\server\tcp>node client
Data: Hello AA00! from localhost:3000
Connection closed

D:\aann\aann-rpt02\server\tcp>
```

Socket으로 전송되는 데이터를 처리하고 종료

6.2.3 tcp server & client : result


파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H)
client.js - 제목 없음(작업 영역) - Visual Studio Code
server.js
client.js
aann > aann-rpt02 > server > tcp > server.js > ...
aann > aann-rpt02 > server > tcp > client.js > ...
문제 출력 디버그 콘솔 터미널
node
cmd
줄 1, 열 1 Tab 크기: 4 UTF-8 CRLF JavaScript kite: ready

```
// tcp server (network server)
var net = require('net');
var port = 3000;

// Network connection using socket
var server = net.createServer(function(socket)
    console.log("Connection from " + socket.re
    socket.end("Hello AA00! from localhost:30
});

server.listen(port, "127.0.0.1");
console.log("Network server started at port :
```

```
// tcp client
var net = require('net');
var port = 3000;
var client = new net.Socket();
// Connection using socket
client.connect(port, "127.0.0.1");
// Receive data from socket
client.on('data', function (data) {
    console.log('Data: ' + data);
    client.destroy();
});

// Add a 'close' event handler for the client
client.on('close', function () {
    console.log('Connection closed');
});
```

```
D:\aann\aann-rpt02\server\tcp>node server
Network server started at port : 3000
Connection from 127.0.0.1
Connection from 127.0.0.1
Connection from 127.0.0.1
Connection from 127.0.0.1
```

```
Data: Hello AA00! from localhost:3000
Connection closed

D:\aann\aann-rpt02\server\tcp>node client
Data: Hello AA00! from localhost:3000
Connection closed

D:\aann\aann-rpt02\server\tcp>node client
Data: Hello AA00! from localhost:3000
Connection closed

D:\aann\aann-rpt02\server\tcp>
```

Save file

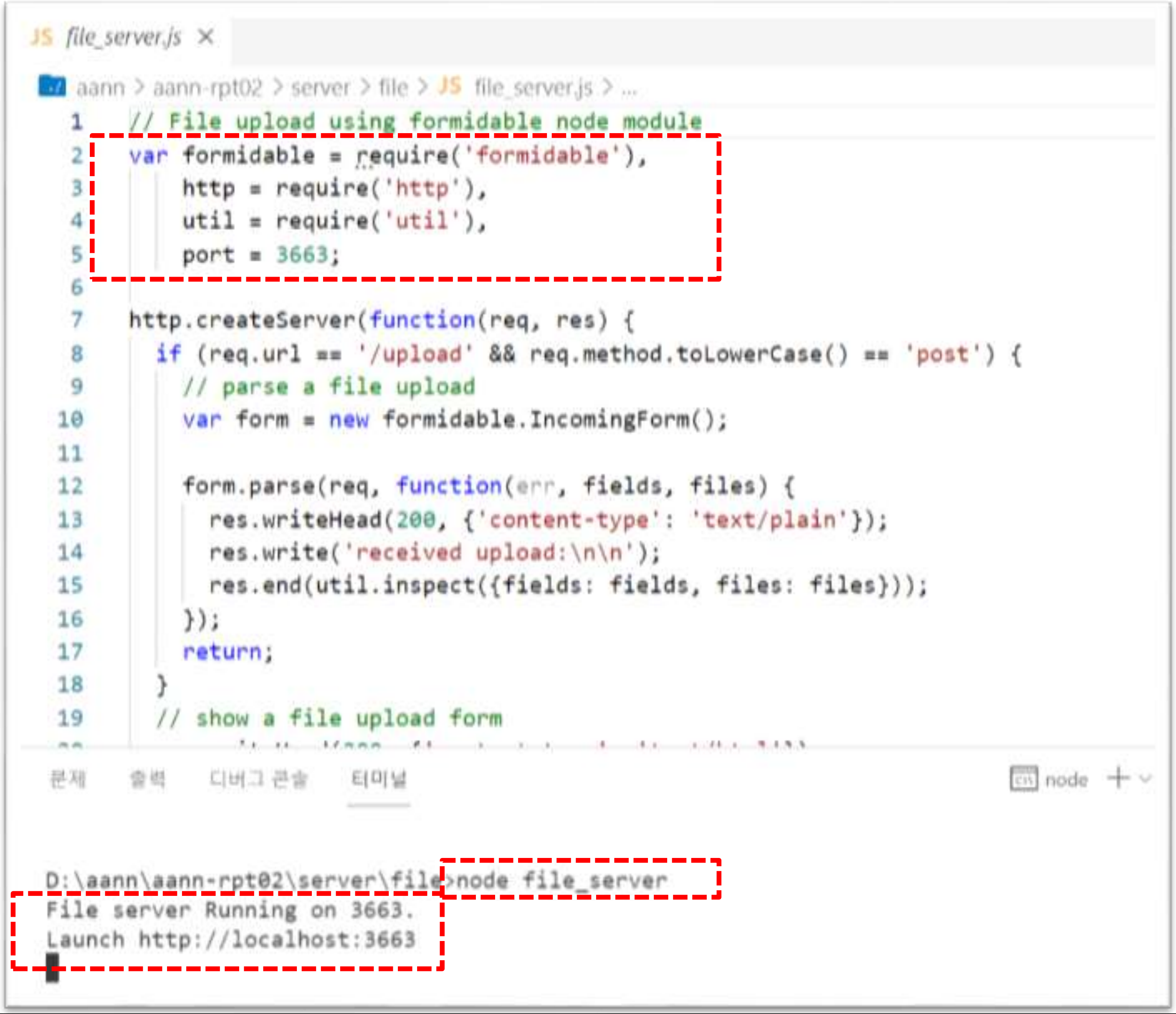
AAnn_TCP_Log.png

6. Node Server

Node Server I.

1. HTTP server
2. TCP server
3. File upload

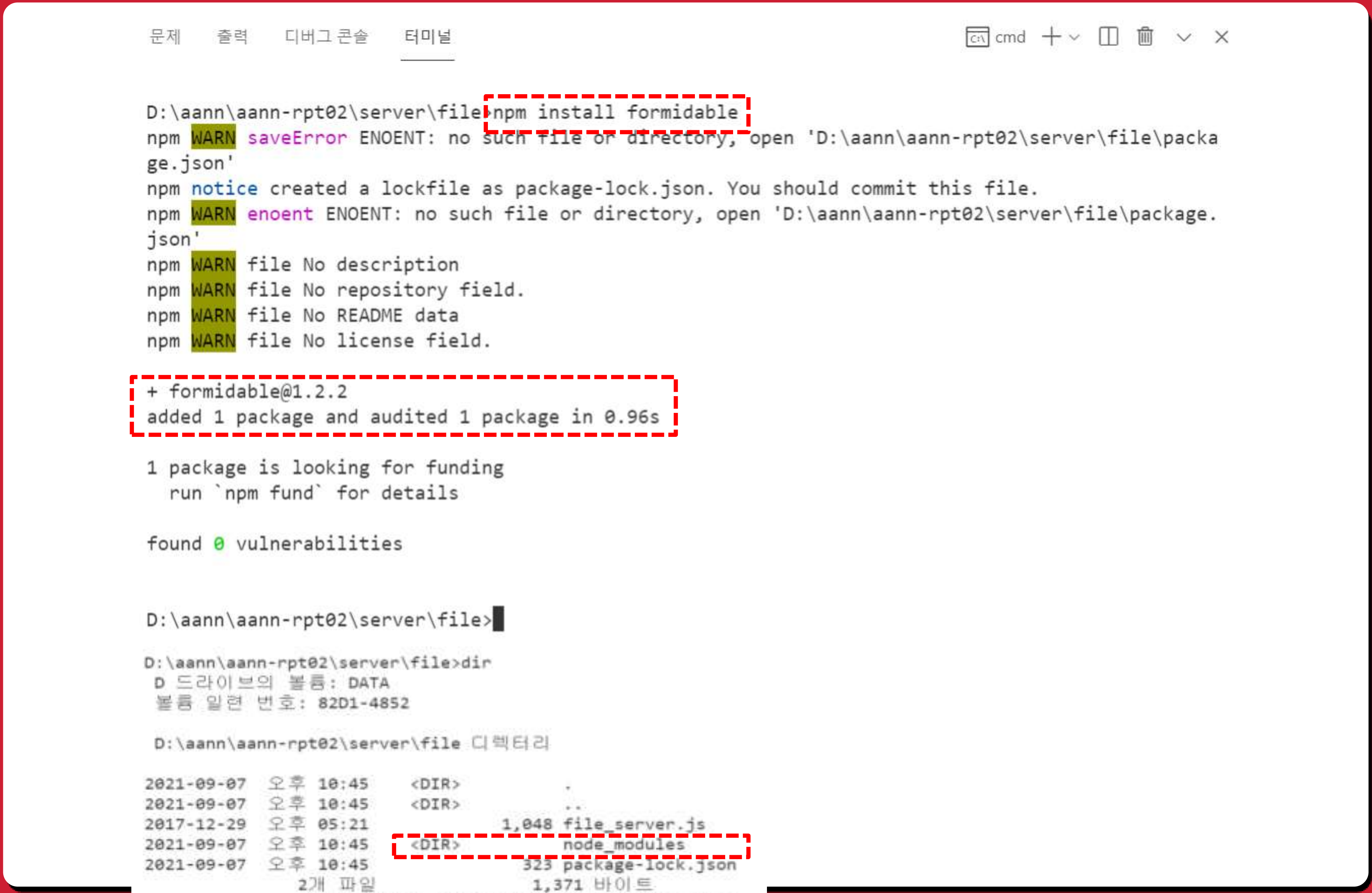
6.3.1 file upload using module 'formidable'


JS file_server.js ×
aann > aann-rpt02 > server > file > JS file_server.js > ...
문제 출력 디버그 콘솔 터미널
node +

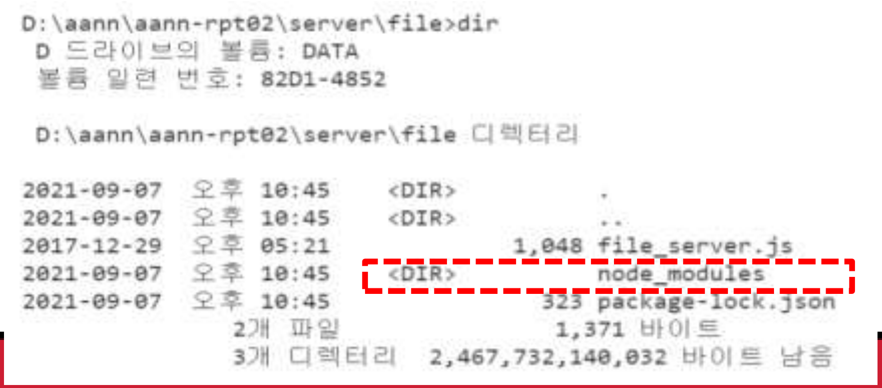
```javascript
// File upload using formidable node module
var formidable = require('formidable'),
    http = require('http'),
    util = require('util'),
    port = 3663;

http.createServer(function(req, res) {
  if (req.url == '/upload' && req.method.toLowerCase() == 'post') {
    // parse a file upload
    var form = new formidable.IncomingForm();

    form.parse(req, function(err, fields, files) {
      res.writeHead(200, {'content-type': 'text/plain'});
      res.write('received upload:\n\n');
      res.end(util.inspect({fields: fields, files: files}));
    });
    return;
  }
  // show a file upload form
```

```
D:\aann\aann-rpt02\server\file>node file_server
File server Running on 3663.
Launch http://localhost:3663
```

6.3.2 file upload : npm install formidable

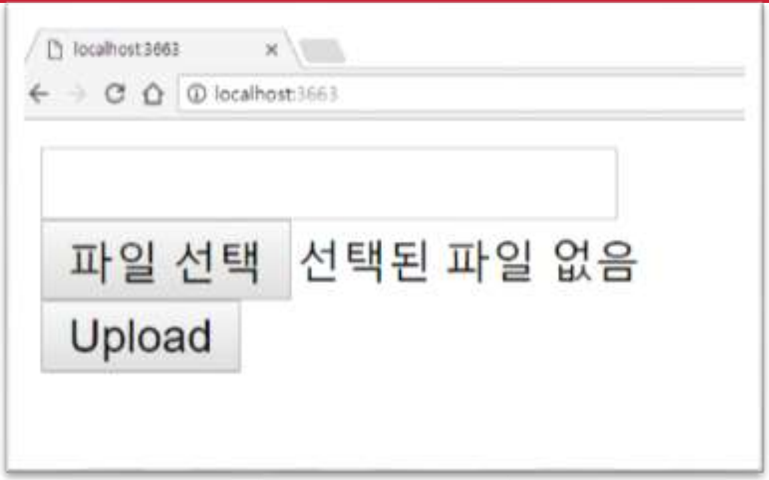


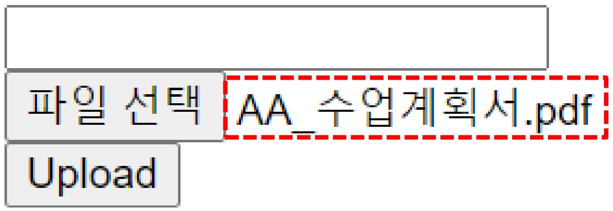
```
D:\aann\aann-rpt02\server\file>npm install formidable
npm WARN saveError ENOENT: no such file or directory, open 'D:\aann\aann-rpt02\server\file\packa
ge.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'D:\aann\aann-rpt02\server\file\package.
json'
npm WARN file No description
npm WARN file No repository field.
npm WARN file No README data
npm WARN file No license field.

+ formidable@1.2.2
added 1 package and audited 1 package in 0.96s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities


D:\aann\aann-rpt02\server\file>
```

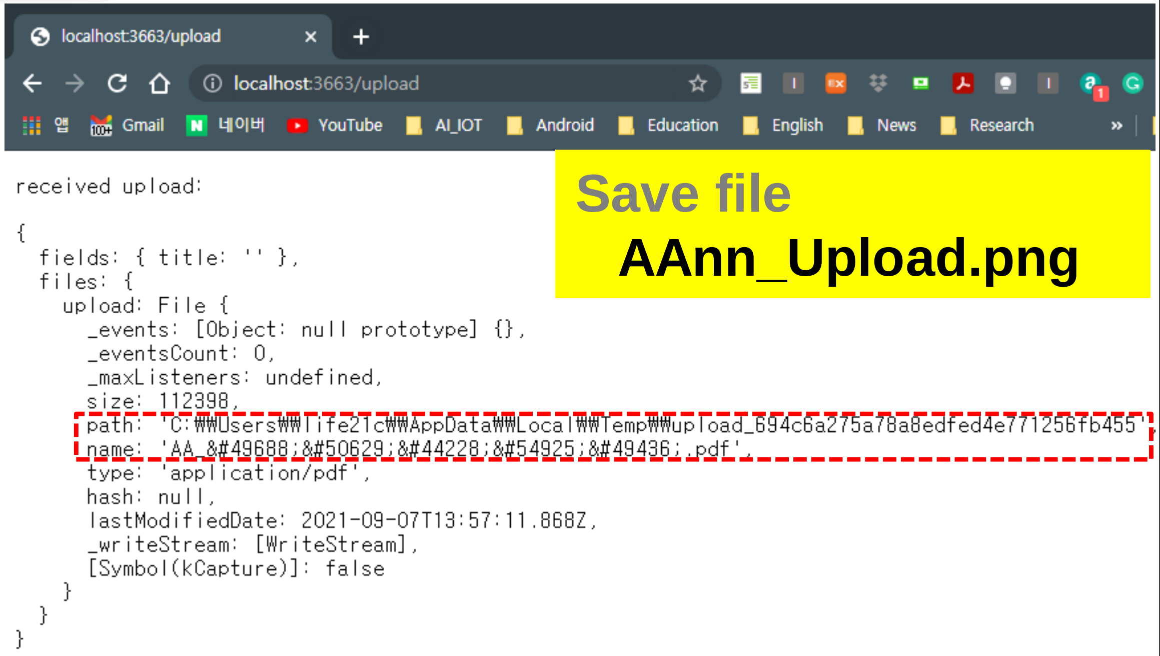


```
D:\aann\aann-rpt02\server\file>dir
 D 드라이브의 볼륨: DATA
 볼륨 일련 번호: 82D1-4852

 D:\aann\aann-rpt02\server\file 디렉터리

2021-09-07  오후 10:45    <DIR>          .
2021-09-07  오후 10:45    <DIR>          ..
2017-12-29  오후 05:21             1,048 file_server.js
2021-09-07  오후 10:45    <DIR>          node_modules
2021-09-07  오후 10:45               323 package-lock.json
               2개 파일               1,371 바이트
               3개 디렉터리  2,467,732,140,032 바이트 남음
```

6.3.3 file upload







```
received upload:

{
  fields: { title: '' },
  files: {
    upload: File {
      _events: [Object: null prototype] {},
      _eventsCount: 0,
      _maxListeners: undefined,
      size: 112398,
      path: 'C:₩₩Users₩₩life21c₩₩AppData₩₩Local₩₩Temp₩₩upload_694c6a275a78a8edfed4e771256fb455',
      name: 'AA_&#49688;&#50629;&#44228;&#54925;&#49436;.pdf',
      type: 'application/pdf',
      hash: null,
      lastModifiedDate: 2021-09-07T13:57:11.868Z,
      _writeStream: [WriteStream],
      [Symbol(kCapture)]: false
    }
  }
}
```

Save file
AAnn_Upload.png

7. Node Server

Node Server II.

1. Express server
2. Full Express App
3. My Express App

7.1.1 Express server test

Step 1 : npm init

Step 2 : npm install --save express

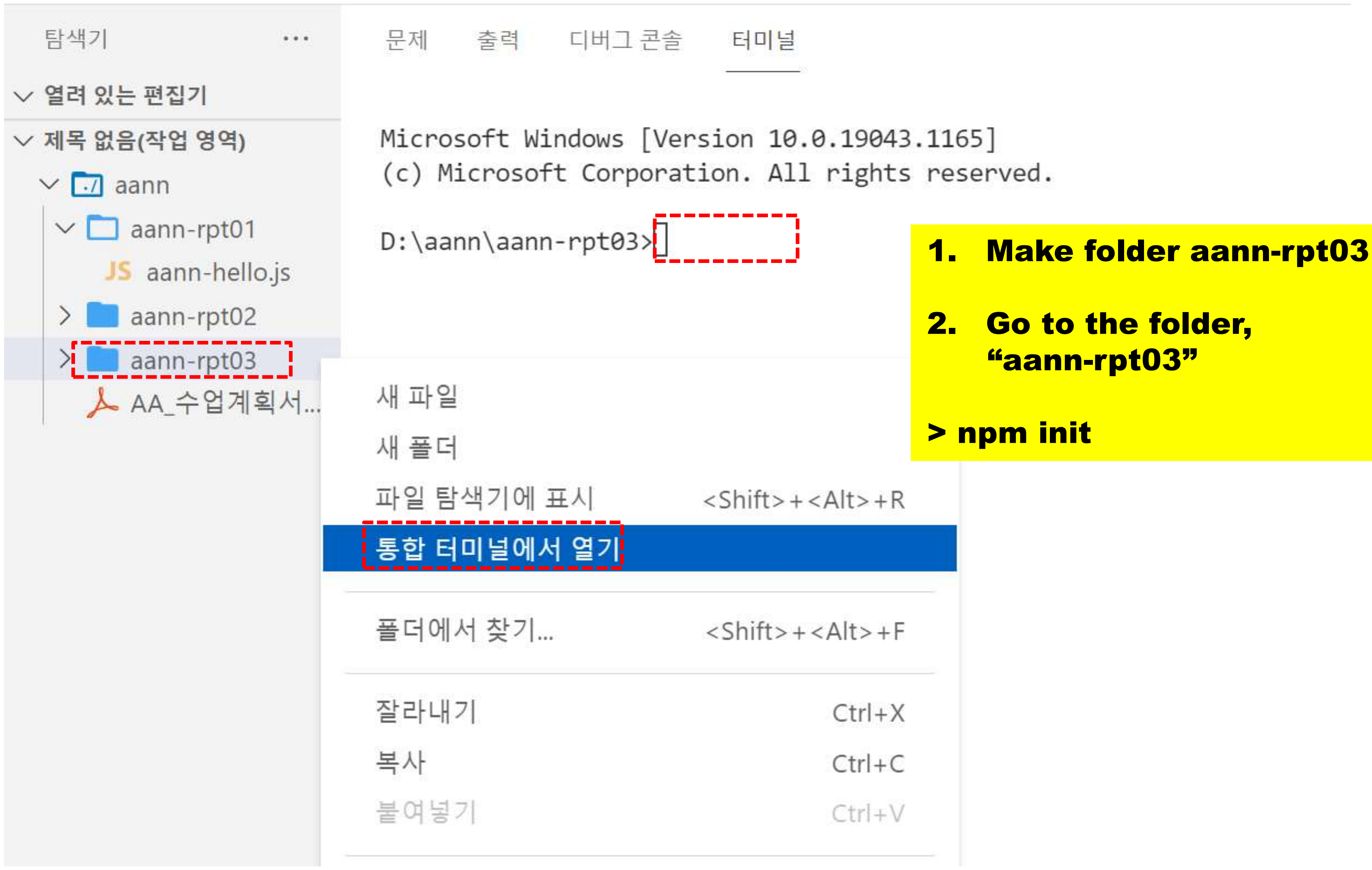
Step 3 : Write Express code

Step 4 : Run app.js

Step 5 : http://localhost:3000

Step 6 : Routing test

7.1.2 Express server test



탐색기 ···

열려 있는 편집기

제목 없음(작업 영역)

aann

aann-rpt01

aann-hello.js

aann-rpt02

aann-rpt03

AA_수업계획서...

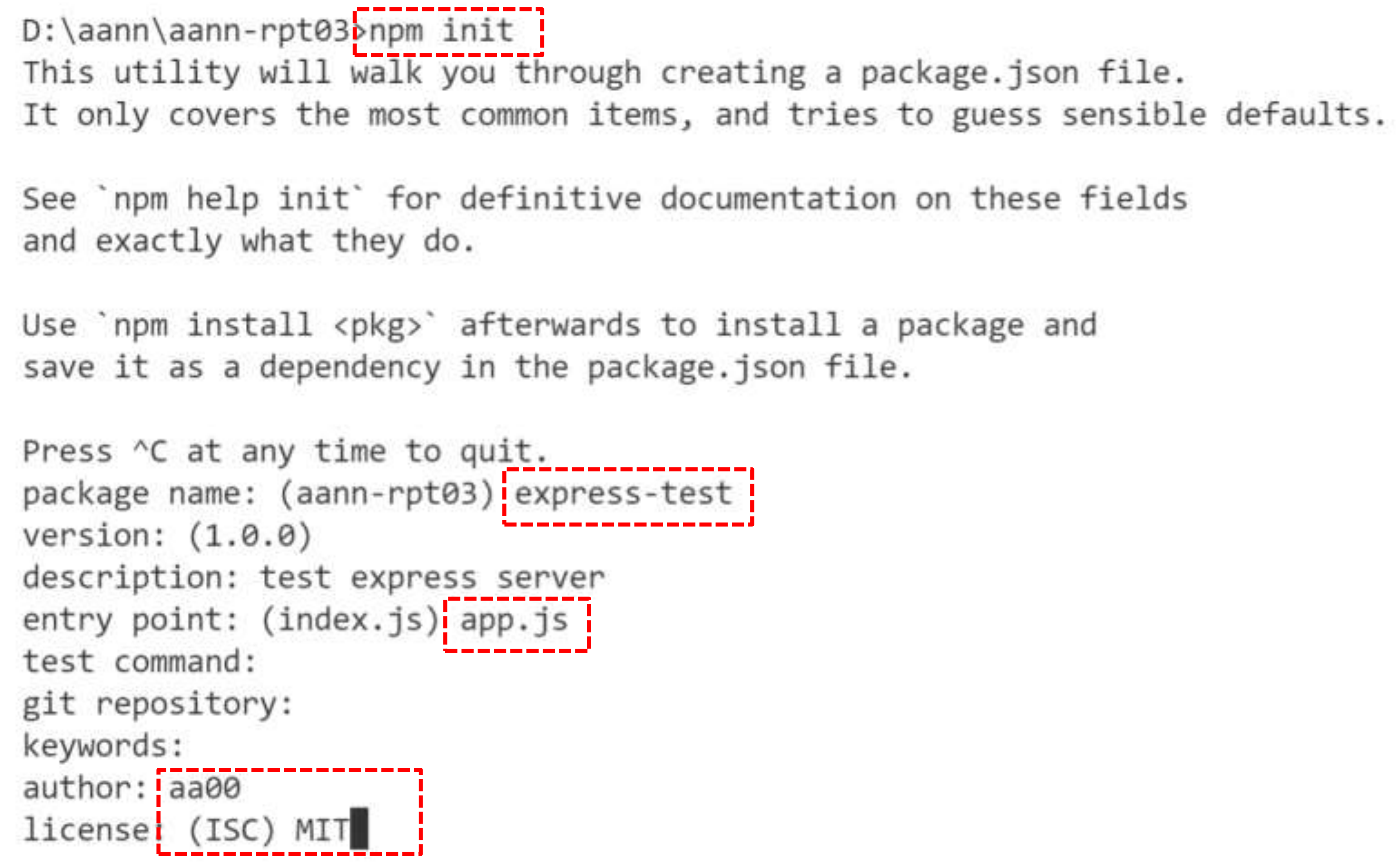
문제 출력 디버그 콘솔 터미널

```
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

D:\aann\aann-rpt03>
```

1. **Make folder aann-rpt03**
2. **Go to the folder, "aann-rpt03"**

> npm init

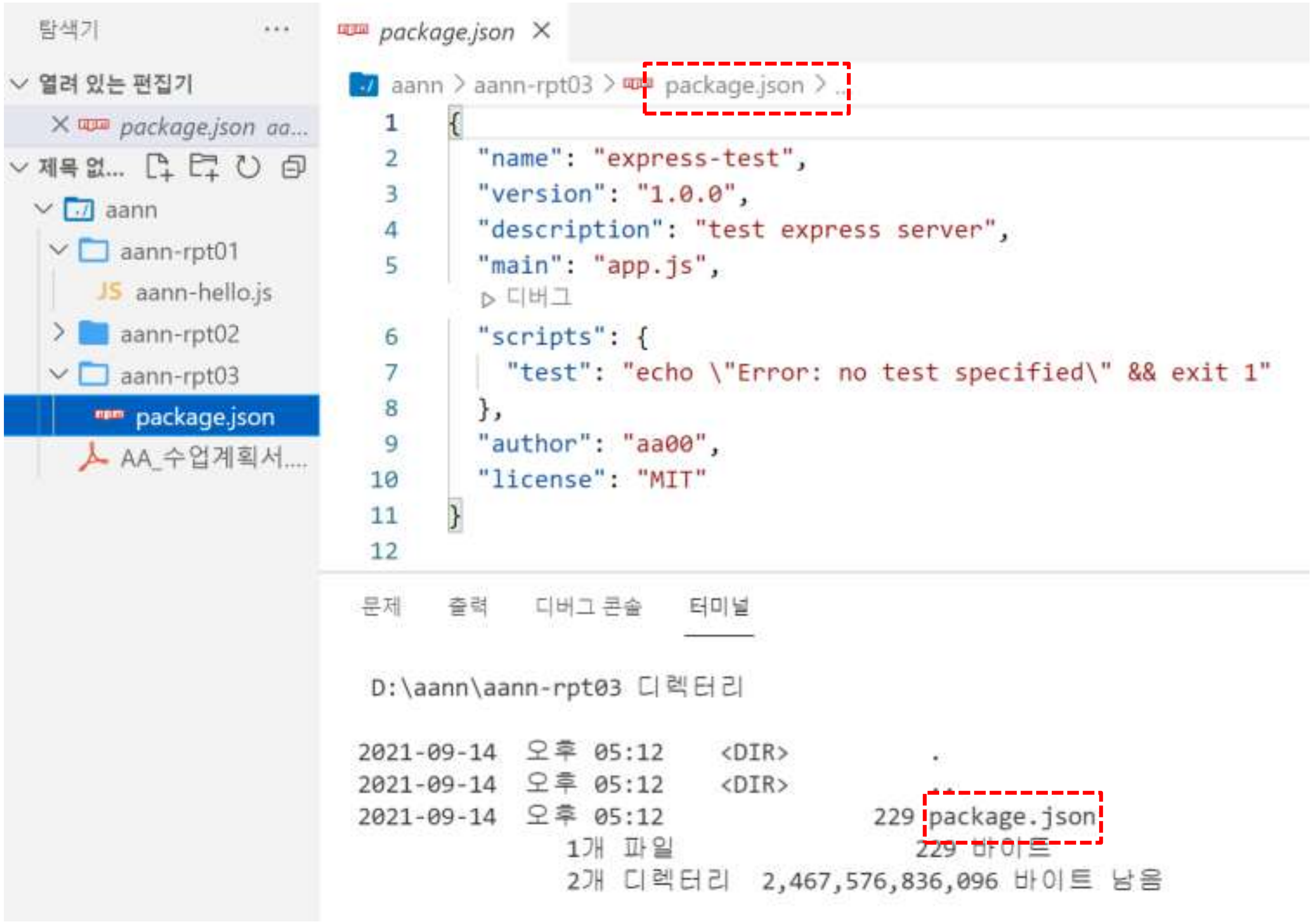
7.1.2 Express server test: npm init



```
D:\aann\aann-rpt03>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: (aann-rpt03) express-test
version: (1.0.0)
description: test express server
entry point: (index.js) app.js
test command:
git repository:
keywords:
author: aa00
license: (ISC) MIT
```

7.1.3 Express server test: package.json

package.json

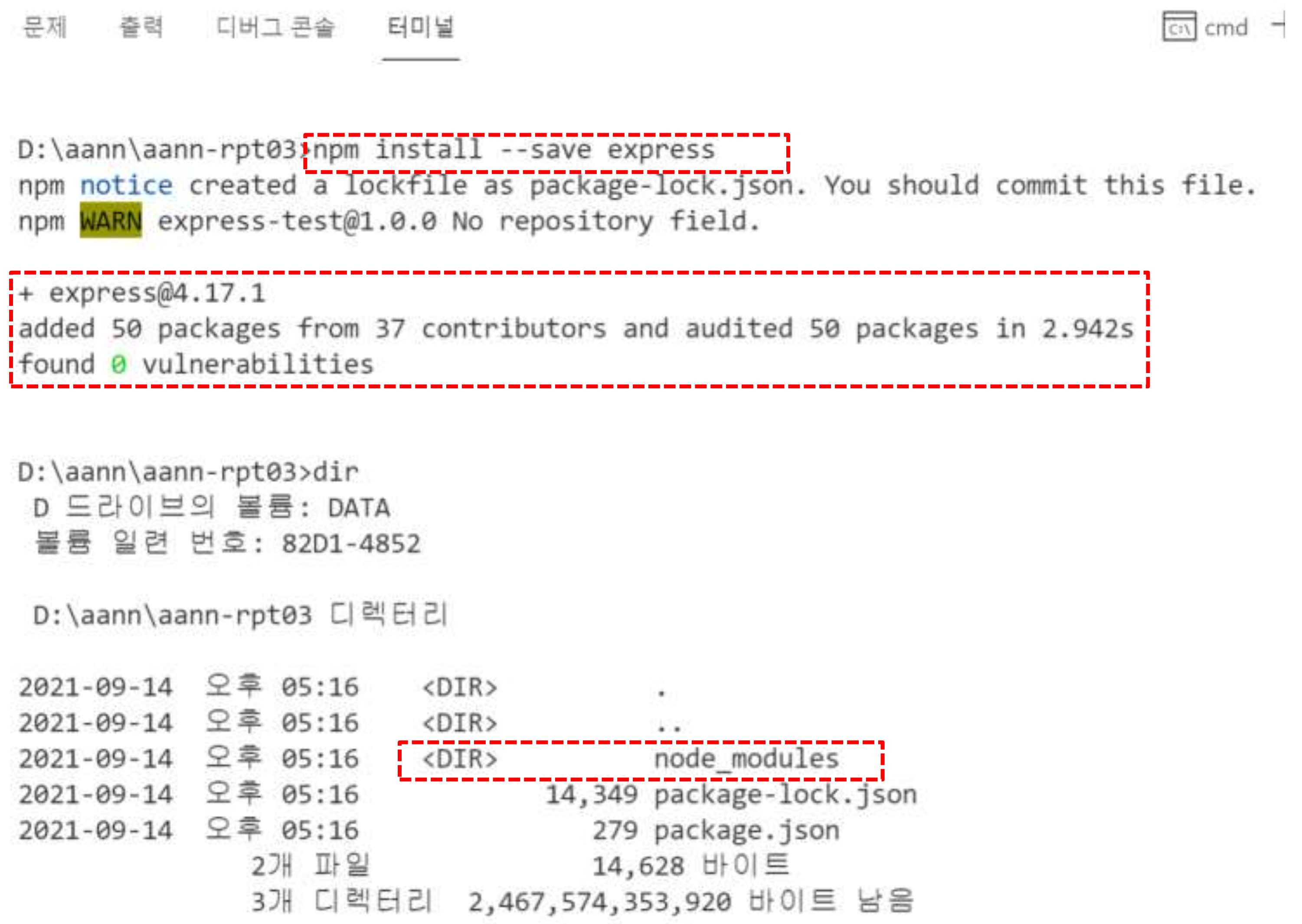


```
{
  "name": "express-test",
  "version": "1.0.0",
  "description": "test express server",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "aa00",
  "license": "MIT"
}
```

```
 D:\aann\aann-rpt03 디렉터리

2021-09-14  오후 05:12    <DIR>          .
2021-09-14  오후 05:12    <DIR>          ..
2021-09-14  오후 05:12               229 package.json
               1개 파일                 229 바이트
               2개 디렉터리  2,467,576,836,096 바이트 남음
```

7.1.4 Express server test: express module install

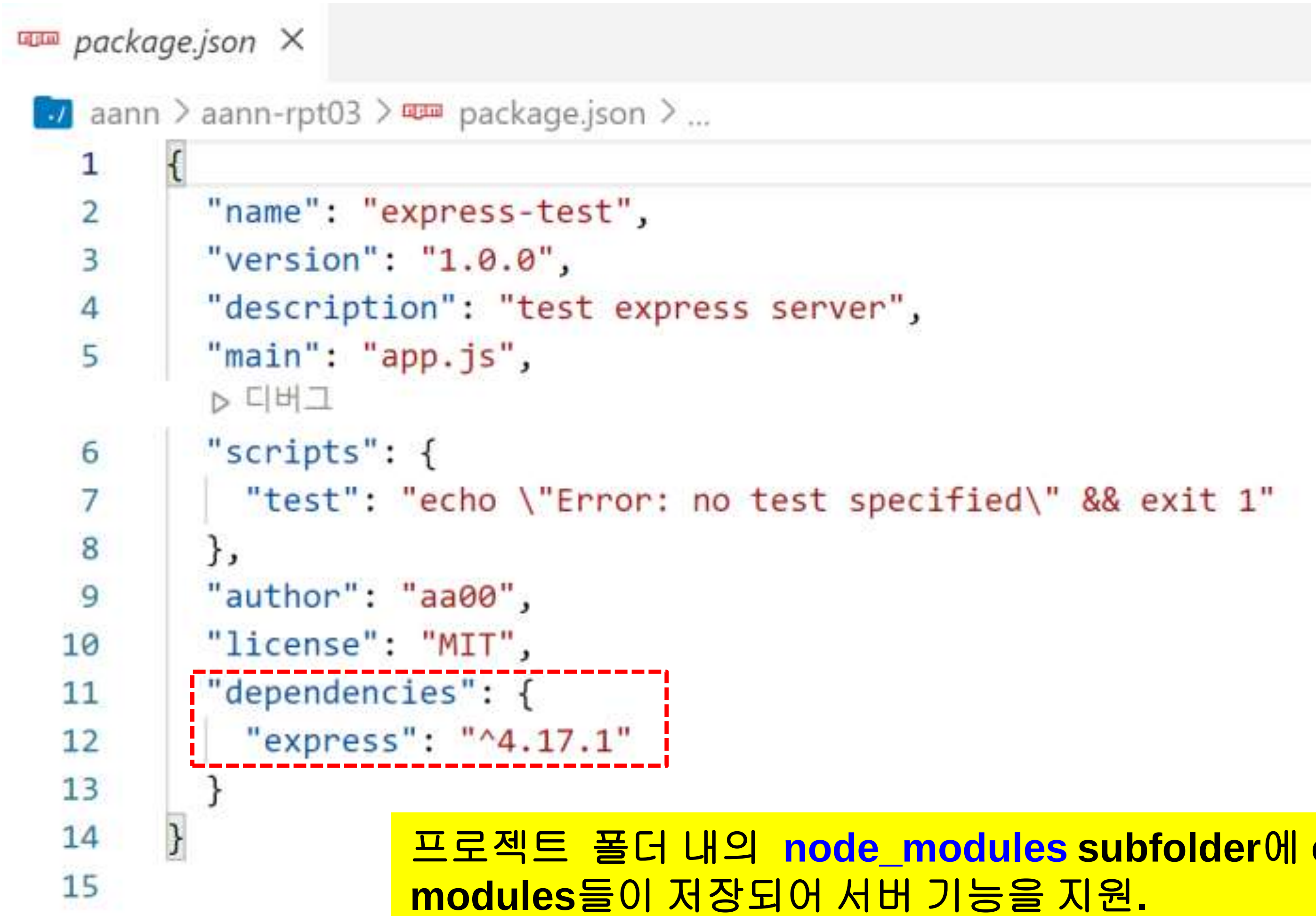
npm install --save express



```
D:\aann\aann-rpt03>npm install --save express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN express-test@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 2.942s
found 0 vulnerabilities


D:\aann\aann-rpt03>dir
 D 드라이브의 볼륨: DATA
 볼륨 일련 번호: 82D1-4852

 D:\aann\aann-rpt03 디렉터리

2021-09-14  오후 05:16    <DIR>          .
2021-09-14  오후 05:16    <DIR>          ..
2021-09-14  오후 05:16    <DIR>          node_modules
2021-09-14  오후 05:16            14,349 package-lock.json
2021-09-14  오후 05:16               279 package.json
               2개 파일              14,628 바이트
               3개 디렉터리  2,467,574,353,920 바이트 남음
```

7.1.5 Express server test: express install

package.json



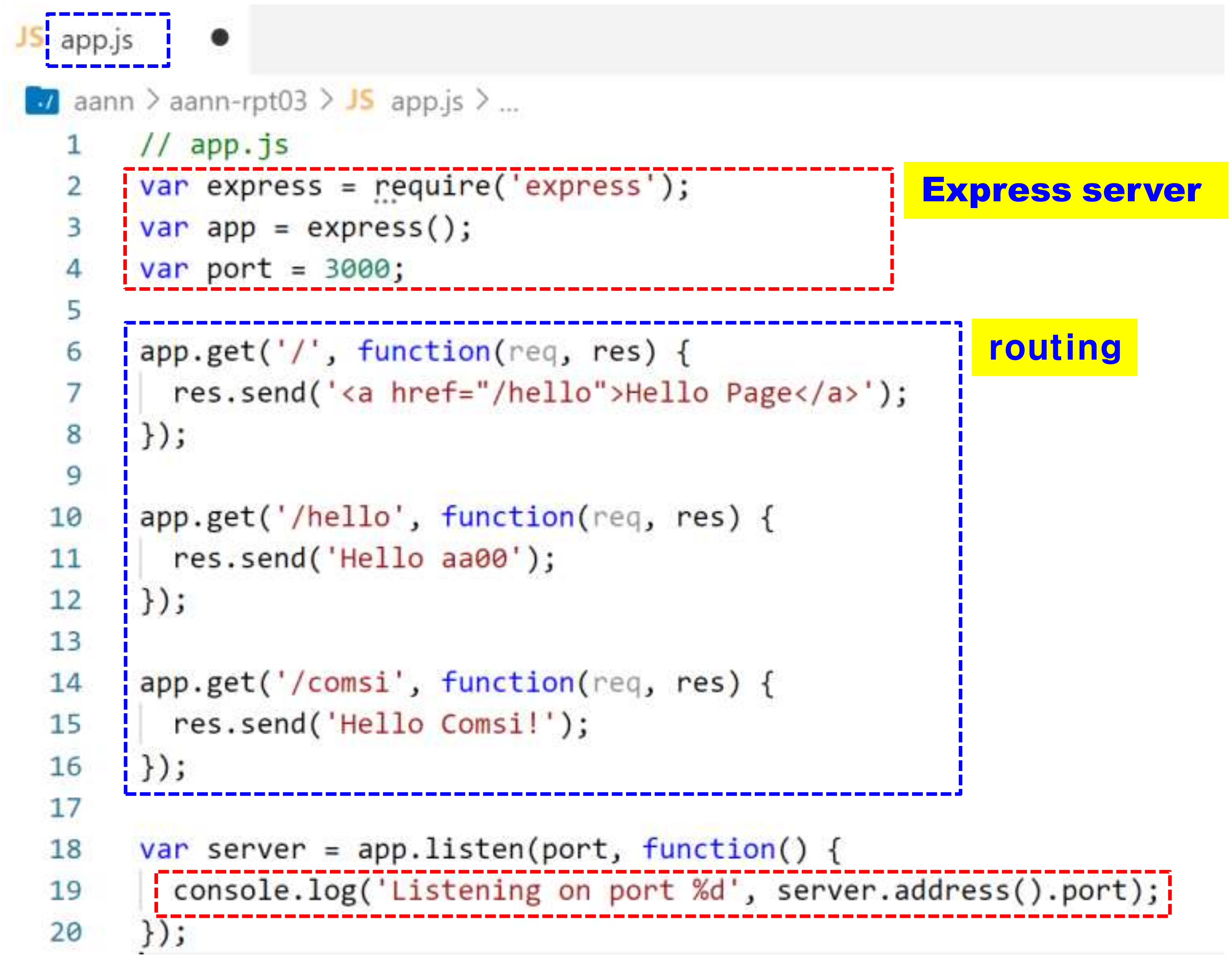
package.json ×

aann > aann-rpt03 > package.json > ...

```
{
  "name": "express-test",
  "version": "1.0.0",
  "description": "test express server",
  "main": "app.js",
  ▷ 디버그
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "aa00",
  "license": "MIT",
  "dependencies": {
    "express": "^4.17.1"
  }
}
```

프로젝트 폴더 내의 **node_modules subfolder**에 **express server modules**들이 저장되어 서버 기능을 지원.
그리고 **package.json**에 **express** 모듈 정보가 **"dependencies"** 속성에 저장.

7.1.6 Express server test: app.js



app.js ●

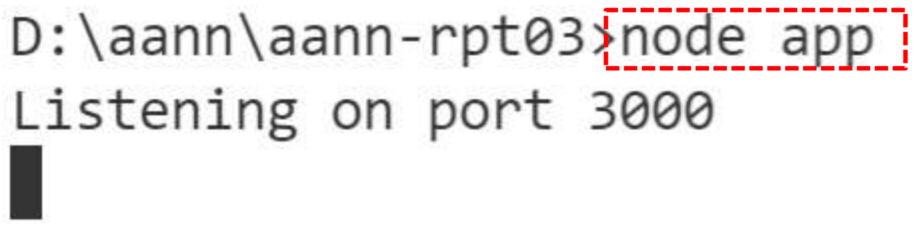
aann > aann-rpt03 > JS app.js > ...

```
// app.js
var express = require('express');
var app = express();
var port = 3000;

app.get('/', function(req, res) {
  res.send('<a href="/hello">Hello Page</a>');
});

app.get('/hello', function(req, res) {
  res.send('Hello aa00');
});

app.get('/comsi', function(req, res) {
  res.send('Hello Comsi!');
});

var server = app.listen(port, function() {
  console.log('Listening on port %d', server.address().port);
});
```

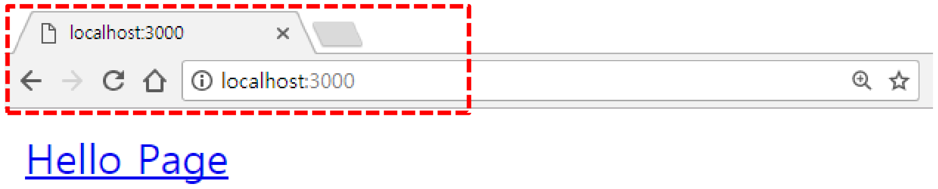
Express server

routing

7.1.7 Express server test: run app.js

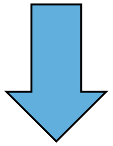


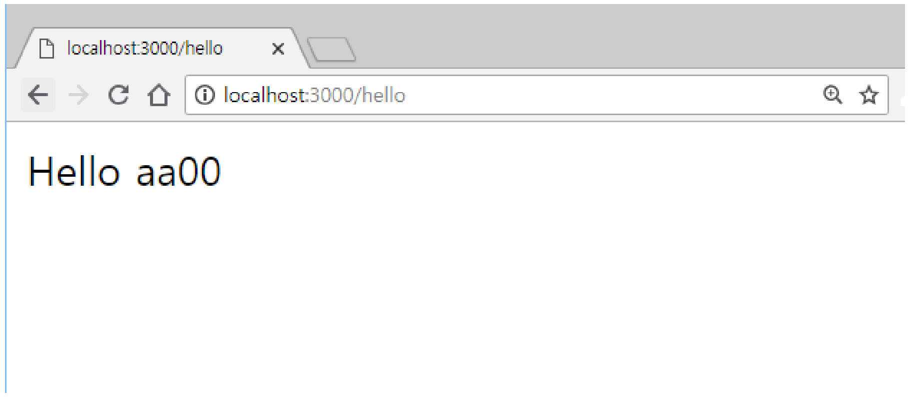
```
D:\aann\aann-rpt03>node app
Listening on port 3000
```



localhost:3000

Hello Page



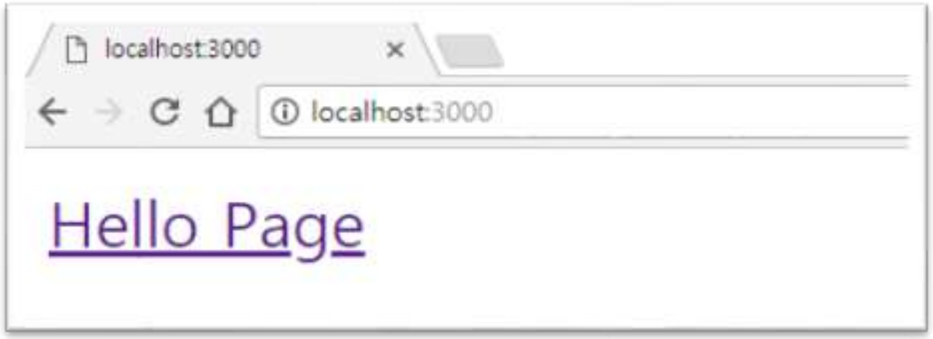


localhost:3000/hello

Hello aa00

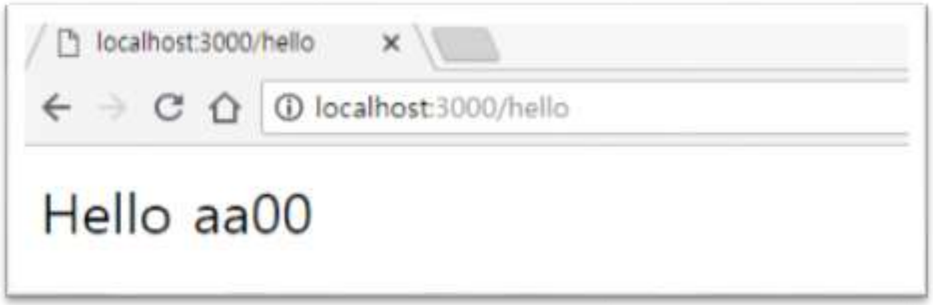
7.1.8 Express server test: test server routing

localhost:3000


localhost:3000

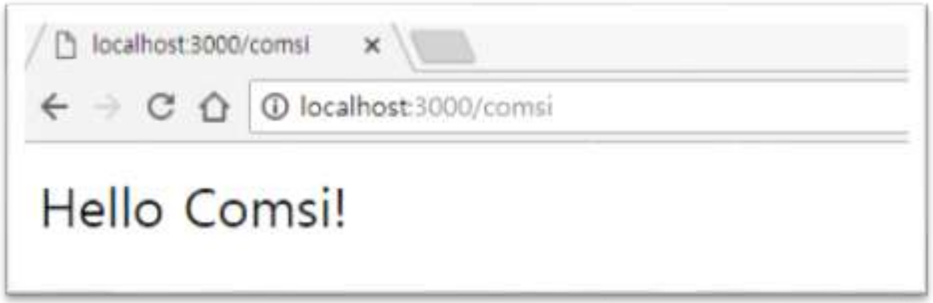
Hello Page

localhost:3000/hello


localhost:3000/hello

Hello aa00

localhost:3000/comsi

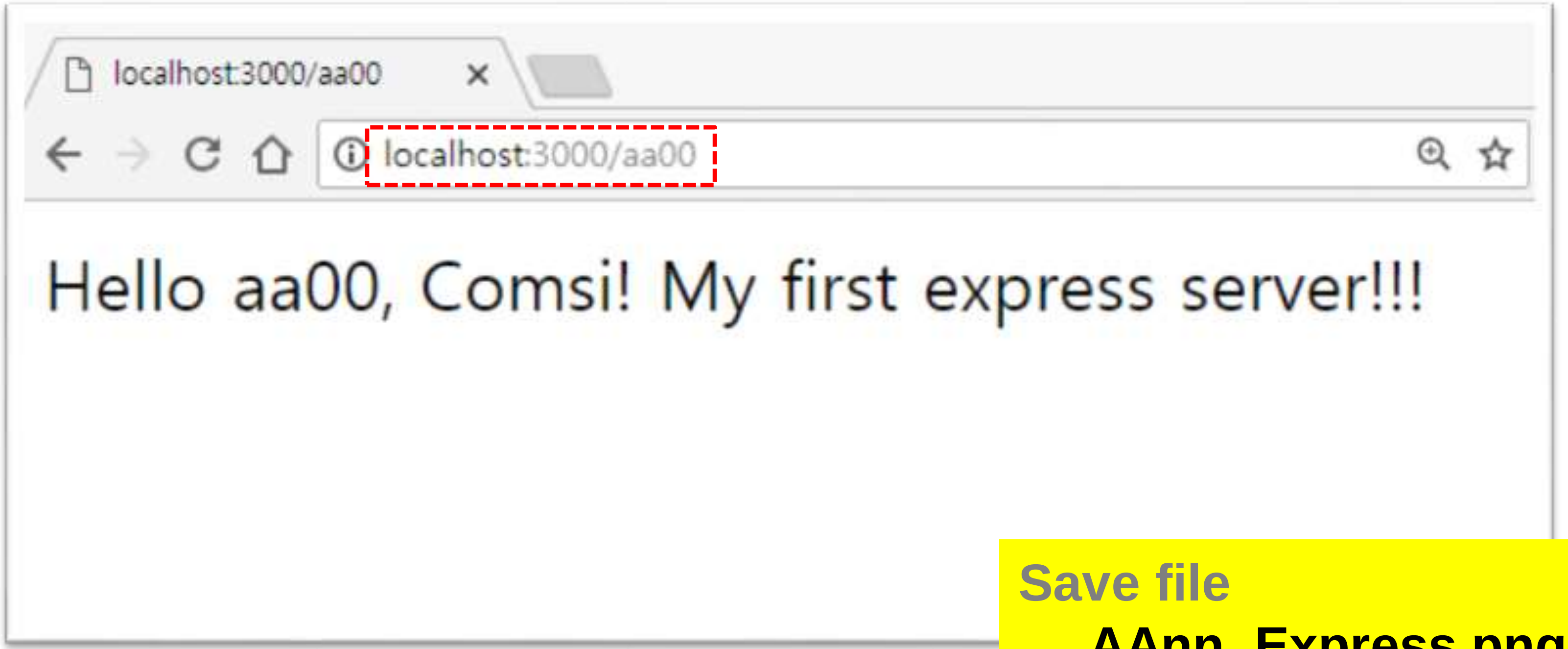

localhost:3000/comsi

Hello Comsi!

7.1.9 Express server test: run app.js - DIY

[DIY] My ID routing → localhost:3000/aann

```
app.get('/aa00', function(req, res) {
  res.send('Hello aa00, Comsi!  My first express server!!!');
});
```



localhost:3000/aa00

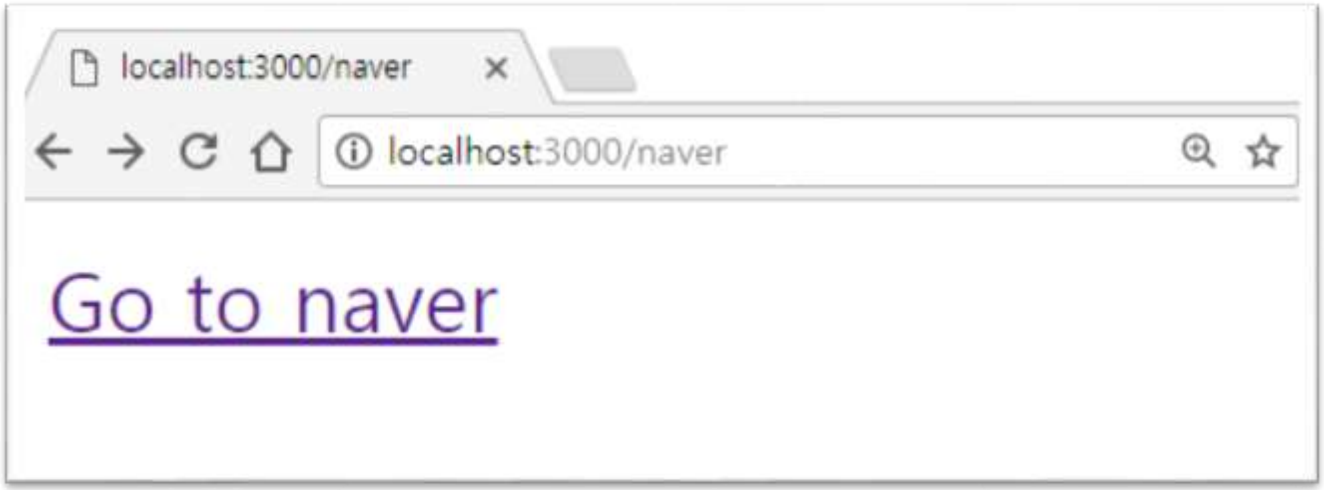
Hello aa00, Comsi! My first express server!!!

Save file
AAnn_Express.png

Routing: **라우팅은 애플리케이션 엔드 포인트(URI)의 정의, 그리고 URI가 클라이언트 요청에 응답하는 방식**

7.1.10 Express server test: run app.js - DIY

[DIY] Go to naver.com

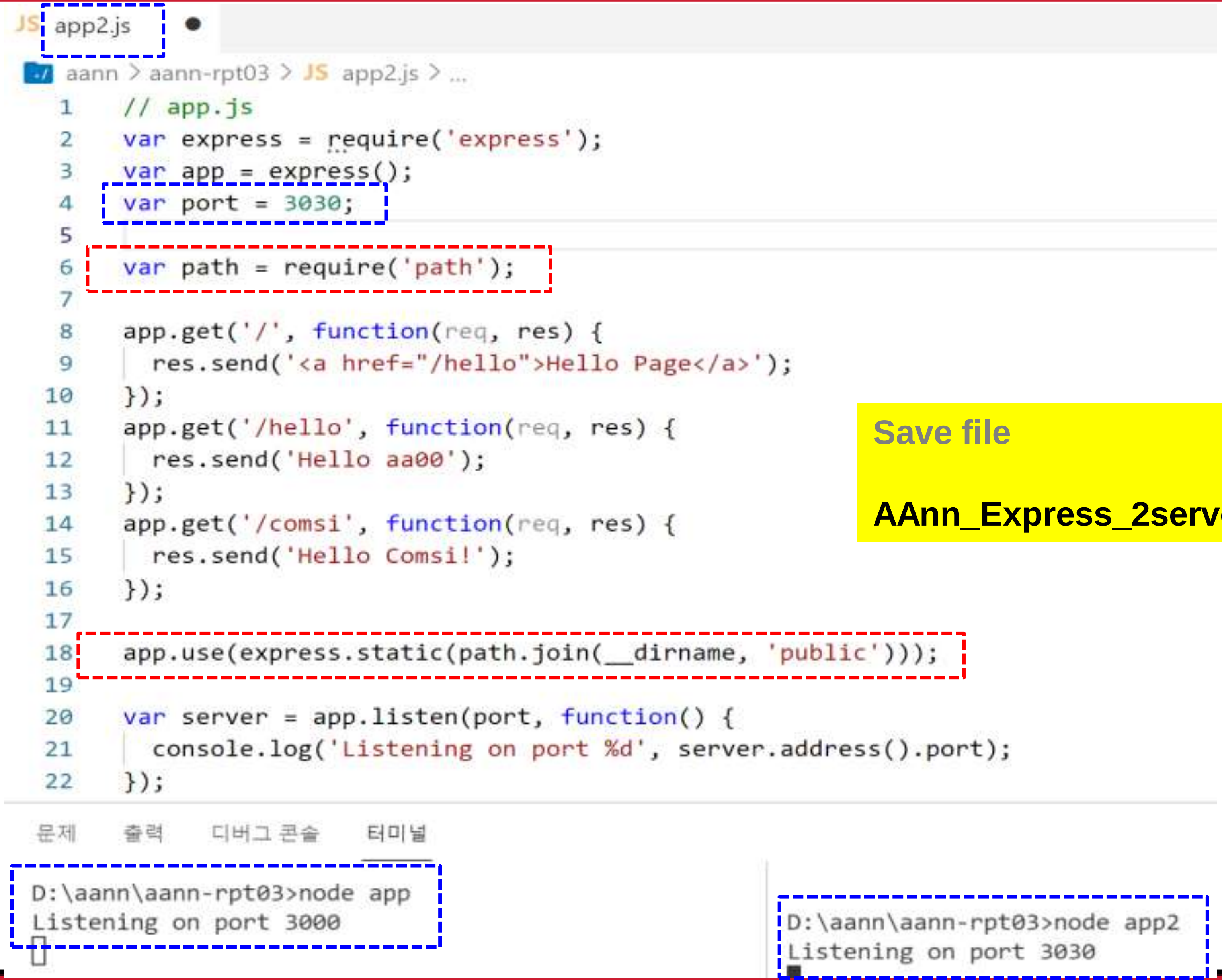

localhost:3000/naver

localhost:3000/naver

Go to naver

[Hint] `<a href="http://www.naver.com">Go to naver</a>`

7.1.12 Express web server test: app2.js



```
// app.js
var express = require('express');
var app = express();
var port = 3030;

var path = require('path');

app.get('/', function(req, res) {
  res.send('<a href="/hello">Hello Page</a>');
});
app.get('/hello', function(req, res) {
  res.send('Hello aa00');
});
app.get('/comsi', function(req, res) {
  res.send('Hello Comsi!');
});

app.use(express.static(path.join(__dirname, 'public')));

var server = app.listen(port, function() {
  console.log('Listening on port %d', server.address().port);
});
```

```
D:\aann\aann-rpt03>node app
Listening on port 3000
```

```
D:\aann\aann-rpt03>node app2
Listening on port 3030
```

Save file

AAnn_Express_2server.png

[Practice]

- [wk03]
  - Express server
  - Upload folder: aann-rpt03
  - Use repo "aann" in github

wk03 : Practice-02 : aann-rpt03

◆ **[Target of this week]**

- Complete your works
- **Save your outcomes and 1 figure**

Upload folder : aann-rpt03

- 제출할 파일들

① **AAnn_Express.png**

② **AAnn_Express_2server.png**

③ **app.js**

④ **app2.js**

Lecture materials

- References & good sites

  - ✓ http://www.arduino.cc Arduino Homepage
  - ✓ http://www.nodejs.org/ko Node.js
  - ✓ https://plot.ly/ plotly
  - ✓ https://www.mongodb.com/ MongoDB
  - ✓ https://www.anaconda.com/ Anaconda
  - ✓ http://www.github.com GitHub
  - ✓ https://colab.research.google.com/ Colab