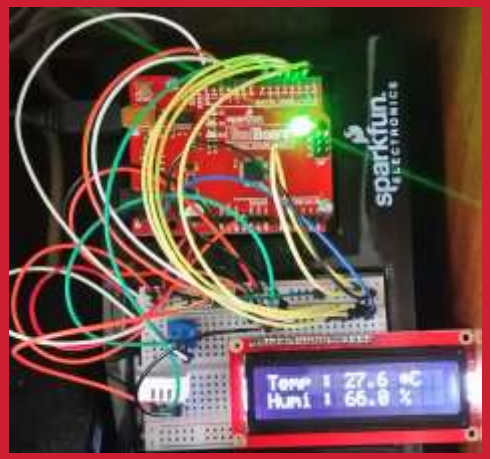


Arduino-IOT

[wk03]

node.js server

Visualization of Signals using Arduino, Node.js & storing signals in MongoDB & mining data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2020

Email : chaos21c@gmail.com

My ID

1분반-목요일 (2학년)

- AA1-01: 강서현
- AA1-02: 강태민
- AA1-03: 김세은
- AA1-04: 여수민
- AA1-05: 정영훈
- AA1-06: 차혁준
- AA1-07: 하태헌
- AA1-08: 김경욱
- AA1-09: 김민욱
- AA1-10: 김민성
- AA1-11: 김민준
- AA1-12: 김인수
- AA1-13: 김현식
- AA1-14: 장성운
- AA1-15: 전승진
- AA1-16: 정희철
- AA1-17: 조동현
- AA1-18: 전동빈
- AA1-19: 신종원

2분반-수요일 (3학년)

- AA2-01: 강민수
- AA2-02: 구병준
- AA2-03: 김종민
- AA2-04: 박성철
- AA2-05: 이승현
- AA2-06: 이창호
- AA2-07: 손성빈
- AA2-08: 안예찬
- AA2-09: 유종인
- AA2-10: 이석민
- AA2-11: 이정문
- AA2-12: 이주원
- AA2-13: 정재영
- AA2-14: 하태성
- AA2-15: 김경미
- AA2-16: 김규년
- AA2-17: 김유빈
- AA2-18: 송다은
- AA2-19: 정주은
- AA2-20: 권준표

[Review]

- [wk02]
  - aax-nn-rpt02
  - node start

1.0 What is node.js?

2.1 Install node.js

다운로드

최신 LTS 버전: 12.18.3 (includes npm 6.14.6)

플랫폼에 맞게 미리 빌드된 Node.js 인스톨러나 소스코드를 다운받아서 바로 개발을 시작하세요.

LTS 대다수 사용자에게 추천	현재 버전 최신 기능
Windows Installer node-v12.18.3-x64.msi	macOS Installer node-v12.18.3.pkg
Source Code node-v12.18.3.tar.gz	

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8

https://nodejs.org/ko/download/

New editor: VScode

Visual Studio Code portable

A source code editor developed by Microsoft for Windows, Linux and macOS.

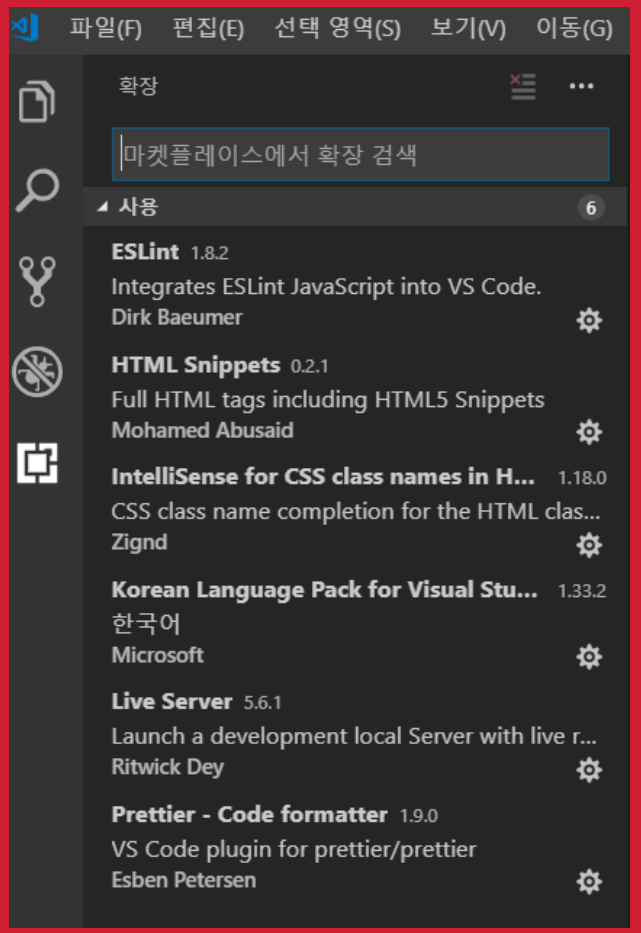
https://portapps.io/app/vscode-portable/

New editor: VScode

확장프로그램 설치 (설치 후 vscode 재시작)

1. Korean language pack
2. HTML snippet
3. IntelliSense for CSS class names in HTML
4. Javascript (ES6)
5. ESLint
6. Prettier
7. Live server (for HTML preview)

8. GitLens, Git History
9. Material Icon Theme
10. Python



C, C++, Java,
Node.js
Python, Jupyter
... all coding!

Node.js Project

npm init

5.2 Node Apps

master aa2-01 / aa2-01-rpt02 /

kangminsooKMS Add files via upload

..

start

AA01_package.PNG

circle.js

circle_info.js

client.js

file_server.js

index.js

server.js

master aa2-01 / aa2-01-rpt02 / start /

kangminsooKMS Add files via upload

..

hello.js

hello_function.js

hello_module.js

hello_my_module.js

package.json

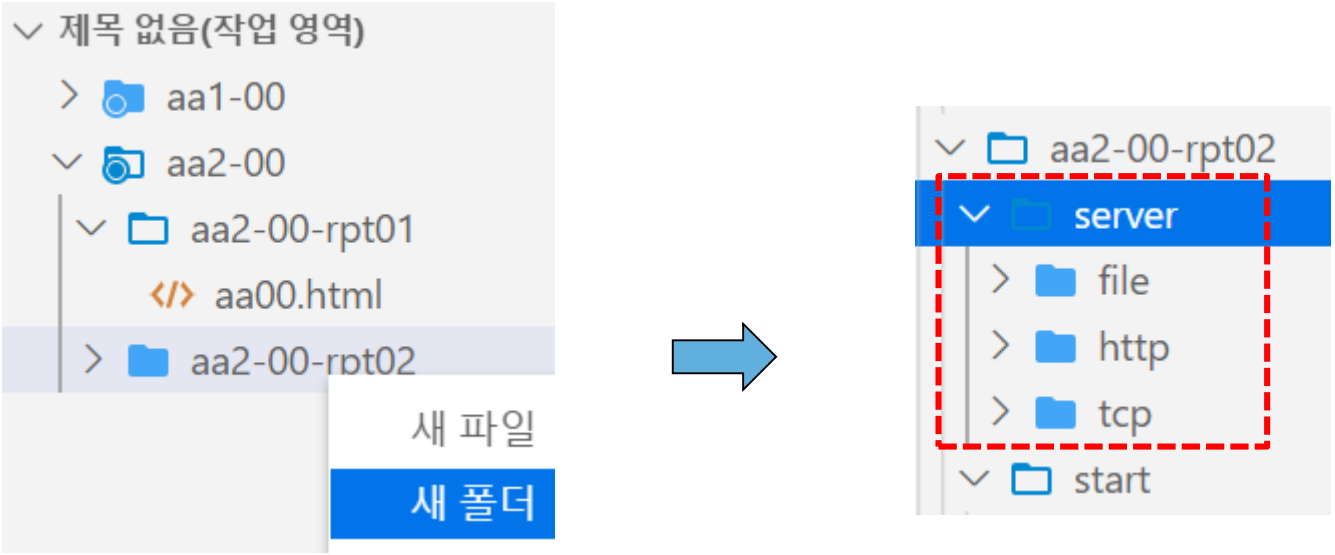
Node.js Server

1. http, tcp, file
2. Express

Node Server I.

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

6.1 Node server : working folders



6.1.1 http server

JS index.js ●

aa2-00 > aa2-00-rpt02 > server > http > JS index.js > ...

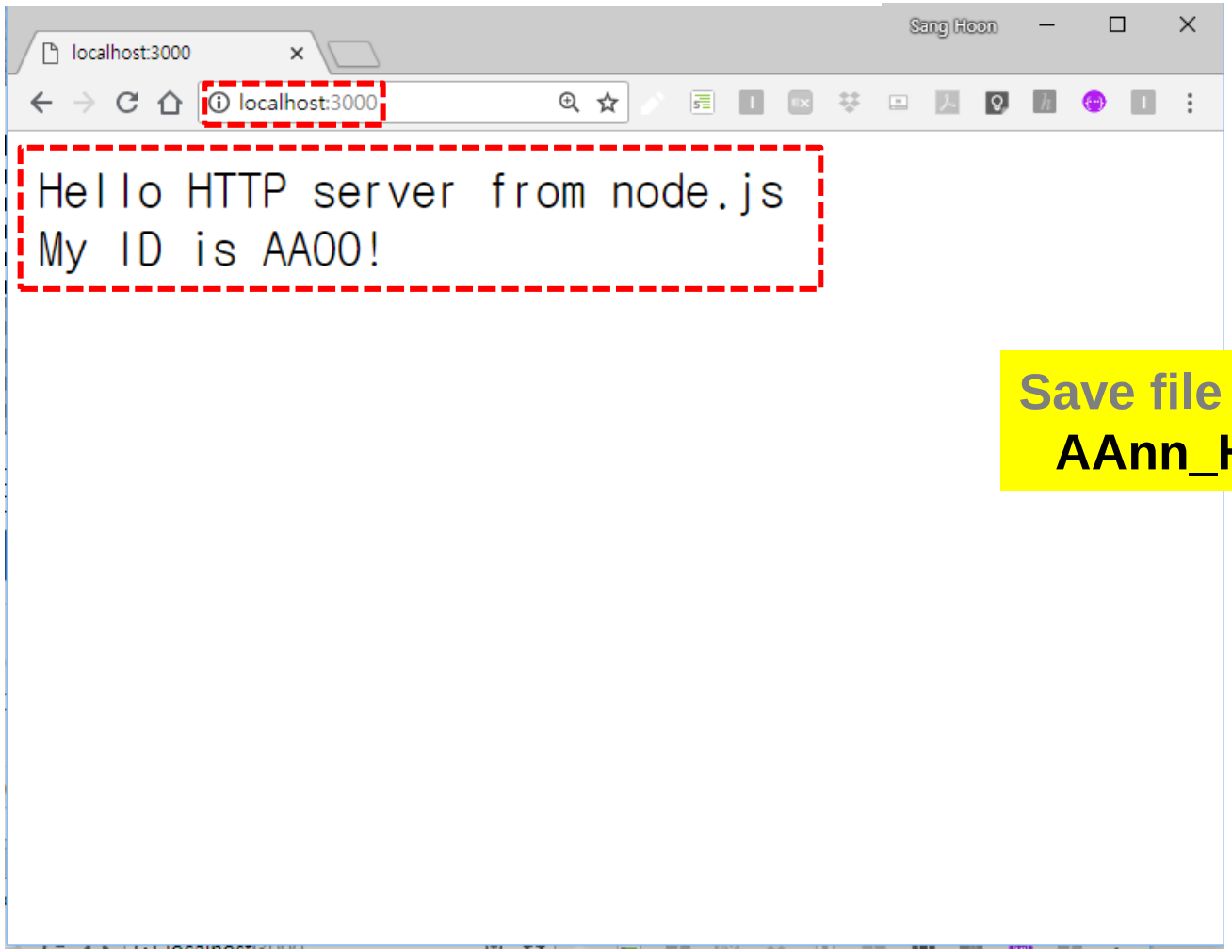
```
1   // http server : index.js
2   var http = require("http");
3   port = 3000;
4
5   var server = http.createServer(function (request, response) {
6     response.writeHeader(200, {
7       "Content-Type": "text/plain",
8     });
9     response.write("Hello HTTP server from node.js"); // WEB response
10    response.write("\n");
11    response.end();
12  });
13
14  server.listen(port);
15  console.log("Server Running on " + port + ".\nLaunch http://localhost:" + port);
```

문제 출력 디버그 콘솔 터미널 1: node

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-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
```



Save file
AAnn_HTTP.png

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

문제 출력 디버그 콘솔 터미널 1: cmd

```
Server Running on 3000.
Launch http://localhost:3000
^C
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\http>
```

[Tip] port number

★사용해도 되는 포트번호와 사용할 수 없는 포트 번호

1. 잘 알려진 포트는 0~1023 까지입니다.

(특정 프로그램들이 사용하기로 예약되어 있기 때문에 쓸 수 없는 포트 번호)

2. 등록된 포트는 1024~49151 까지입니다. (사용가능)

3. 동적 및/또는 개인 포트는 49152~65535 까지입니다. (사용가능)

참고: http://support.microsoft.com/kb/174904/ko

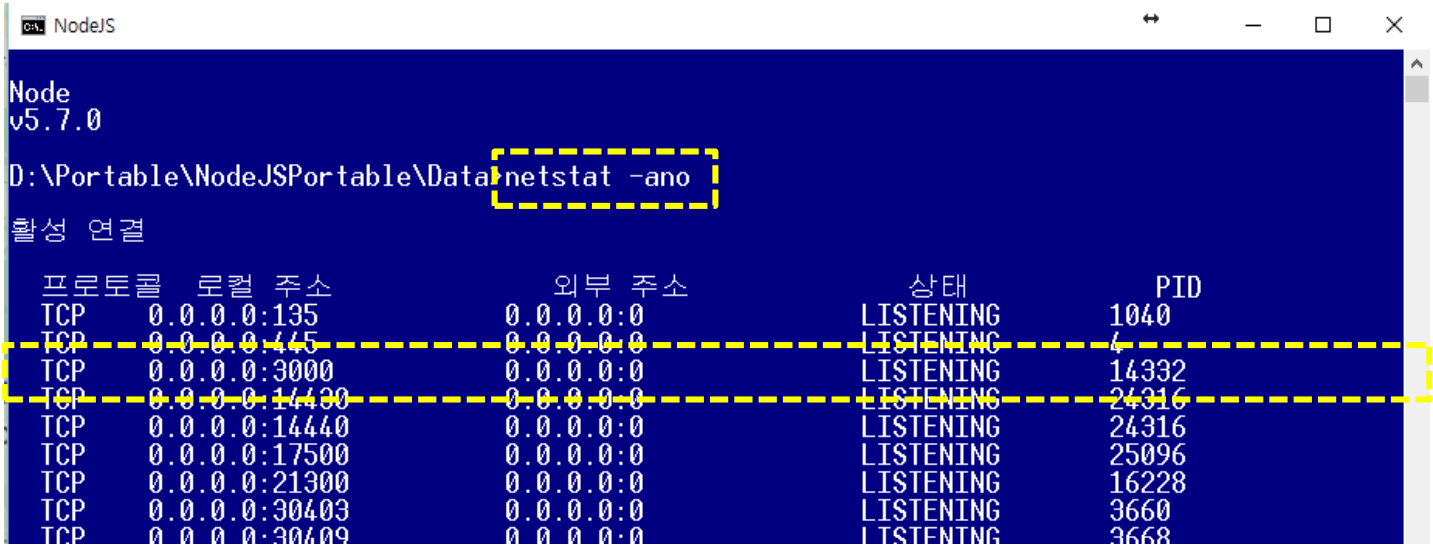
[Tip] listen EADDRINUSE 오류 해결 -1

1. listen **EADDRINUSE 오류**
   -사용중인 포트이거나 포트를 중지시키지 않고 종료시켰을 경우 계속 포트가 사용되고 있는데 연결 시키려고 할 때 나타나는 오류

```
Server Running on 3000.
Launch http://localhost:3000
events.js:154
      throw er; // Unhandled 'error' event
      ^

Error: listen EADDRINUSE :::3000
    at Object.exports._errnoException (util.js:856:11)
    at exports._exceptionWithHostPort (util.js:879:20)
```

2. 해결 방법
   (1) cmd창에서 **netstat –ano**를 입력한 후, 로컬주소에서 사용 중인 포트 번호를 확인
   (2) 사용중인 포트번호를 확인하고 그 해당포트의 **pid번호를 확인**한다.

[Tip] listen EADDRINUSE 오류 해결 -2



2. 해결 방법 (cmd에서 다음 명령 실행 후 port 3000의 pid가 제거됨을 확인)
   taskkill /pid PID_number

```
D:\Portable\NodeJSPortable\Data>taskkill /pid 14332
성공: 프로세스(PID 14332)에 종료 신호를 보냈습니다.
```

```
D:\Portable\NodeJSPortable\Data>netstat -ano

활성 연결

  프로토콜  로컬 주소          외부 주소          상태           PID
  TCP    0.0.0.0:135      0.0.0.0:0      LISTENING       1040
  TCP    0.0.0.0:445      0.0.0.0:0      LISTENING       4
  TCP    0.0.0.0:14430    0.0.0.0:0      LISTENING       24316
```

6.2.1 tcp server (socket connection)

server.js

aa2-00 > aa2-00-rpt02 > server > tcp > server.js > ...

```
1   // tcp server (network server)
2   var net = require("net");
3   var port = 3000;
4
5   // Network connection using socket
6   var server = net.createServer(function (socket) {
7     console.log("Connection from " + socket.remoteAddress);
8     socket.end("Hello AA00! from localhost:3000");
9   });
10
11  server.listen(port, "127.0.0.1");
12  console.log("Network server started at port : " + port);
13
```

Socket으로 전송

문제 출력 디버그 콘솔 터미널 1: node

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>node server
Network server started at port : 3000
```

6.2.2 tcp client

client.js

aa2-00 > aa2-00-rpt02 > server > tcp > client.js > ...

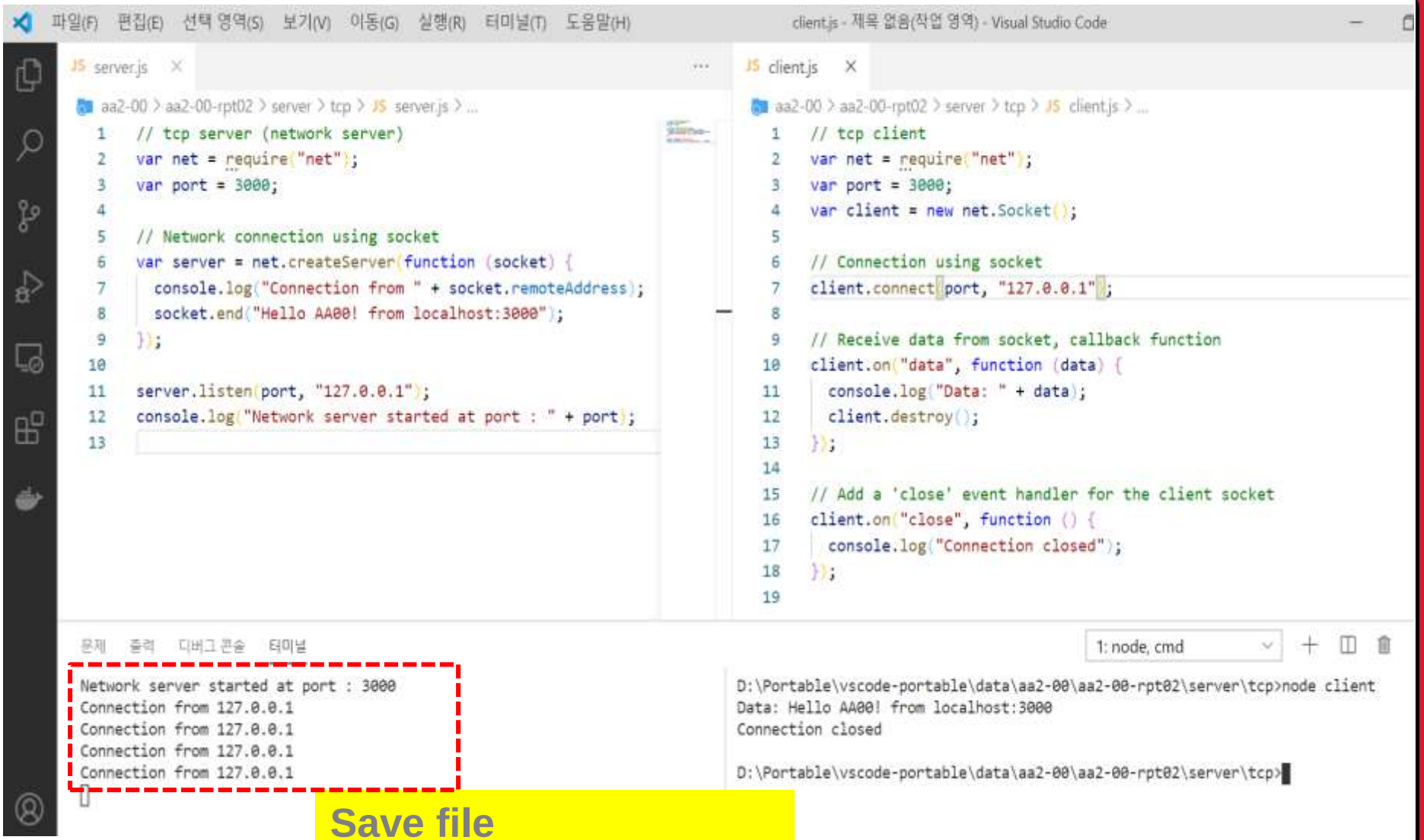
```
1   // tcp client
2   var net = require("net");
3   var port = 3000;
4   var client = new net.Socket();
5
6   // Connection using socket
7   client.connect(port, "127.0.0.1");
8
9   // Receive data from socket, callback function
10  client.on("data", function (data) {
11    console.log("Data: " + data);
12    client.destroy();
13  });
14
15  // Add a 'close' event handler for the client socket
16  client.on("close", function () {
17    console.log("Connection closed");
18  });
19
```

Socket으로 전송되는
데이터를 처리하고 종료

1: node, cmd

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>node client
Data: Hello AA00! from localhost:3000
Connection closed

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\tcp>
```

6.2.3 tcp server & client : result



Save file

AAnn_TCP_Log.png

6.3.1 file upload using module 'formidable'

JS file_server.js ×

aa2-00 > aa2-00-rpt02 > server > file > JS file_server.js > http.createServer() callback > form.pars

```
1  // File upload using formidable node module
2  var formidable = require("formidable"),
3      http = require("http"),
4      util = require("util"),
5      port = 3663;
6
7  http
8    .createServer(function (req, res) {
9      if (req.url == "/upload" && req.method.toLowerCase() == "post") {
10       // parse a file upload
11       var form = new formidable.IncomingForm();
12
13       form.parse(req, function (err, fields, files) {
14         res.writeHead(200, { "content-type": "text/plain" });
15         res.write("received upload:\n\n");
16         res.end(util.inspect({ fields: fields, files: files }));
17       });
18       return;
19     }
```

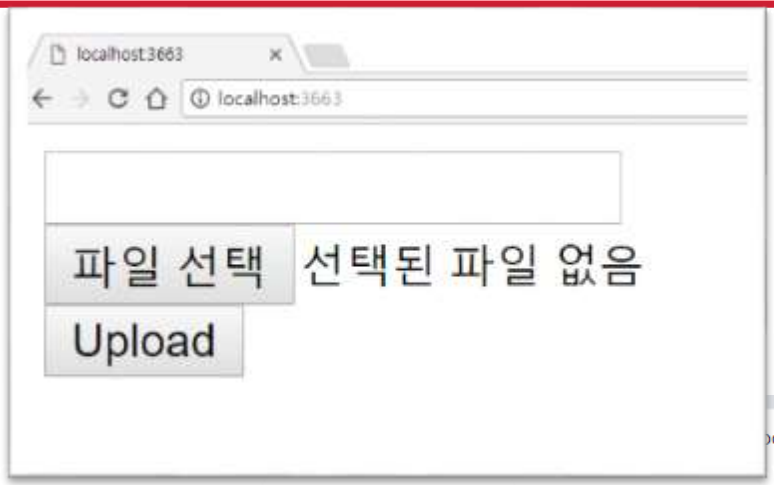
문제 출력 디버그 콘솔 터미널

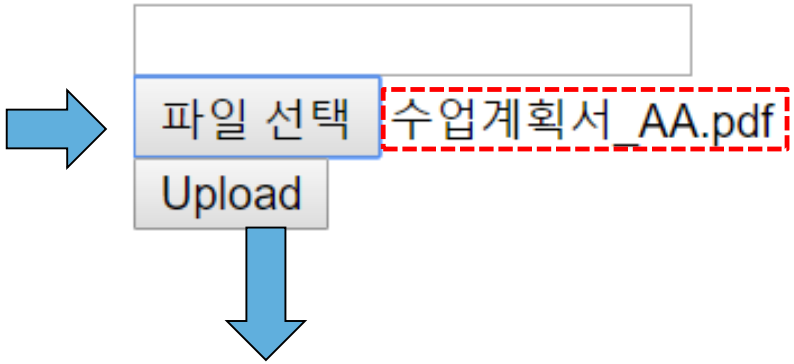
```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file>node file_server
File server Running on 3663.
Launch http://localhost:3663
```

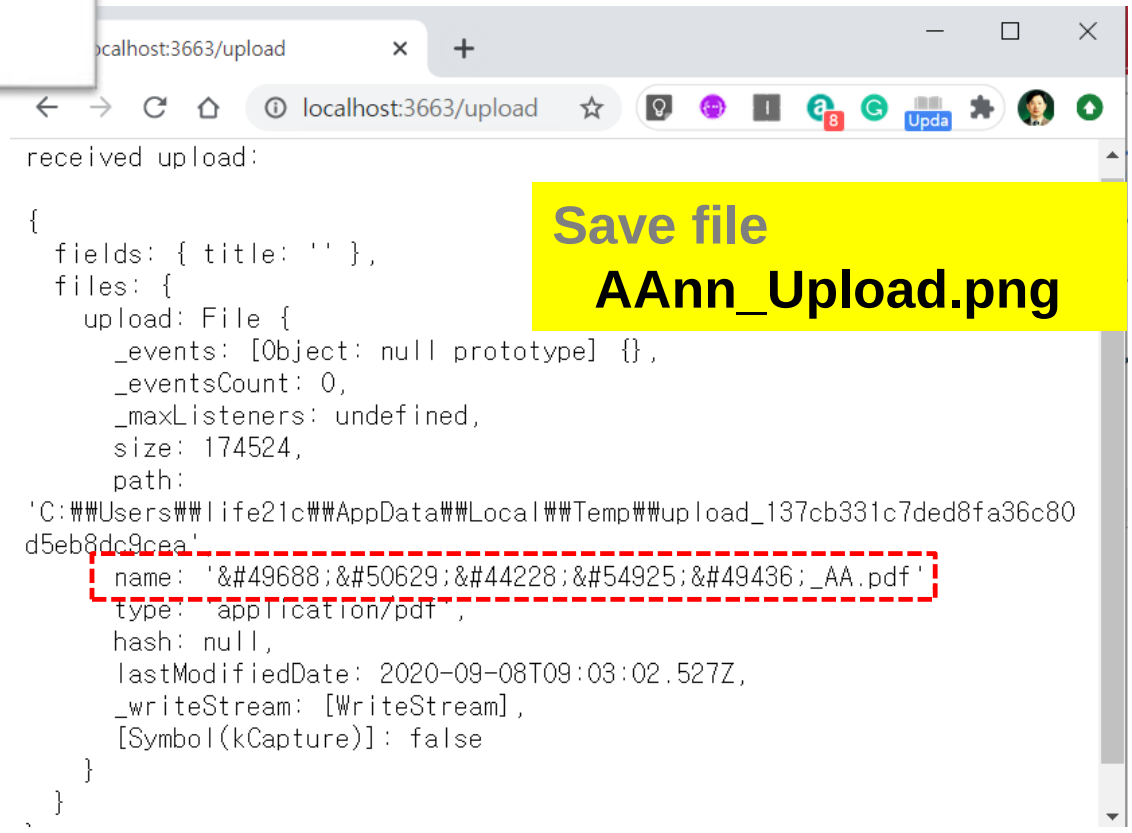
6.3.2 file upload : npm install formidable

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file>npm install formidable
npm WARN saveError ENOENT: no such file or directory, open 'D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file\package.j
son'
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:\Portable\vscode-portable\data\aa2-00\aa2-00-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 0.41s

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

found 0 vulnerabilities


D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file>dir
 D 드라이브의 볼륨: DATA
 볼륨 일련 번호: 82D1-4852

 D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\server\file 디렉터리

2020-09-08  오후 05:56    <DIR>          .
2020-09-08  오후 05:56    <DIR>          ..
2020-09-08  오후 05:54             1,116 file_server.js
2020-09-08  오후 05:56    <DIR>          node_modules
2020-09-08  오후 05:56               323 package-lock.json
               2개 파일               1,439 바이트
               3개 디렉터리  2,515,992,223,744 바이트 남음
```

6.3.3 file upload







[Practice]

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

[practice] local module : aanninfo.js

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

aa2-00 > aa2-00-rpt02 > start > JS AA00info.js > <unknown> > exports

```
1   // AA00info.js
2   module.exports = function (id, name, phone) {
3     console.log("My Info");
4     console.log("ID : " + id);
5     console.log("Name : " + name);
6     console.log("Phone : " + phone + "\n");
7   };
8
```

JS index_AA00.js ×

aa2-00 > aa2-00-rpt02 > start > JS index_AA00.js > ...

```
1   // index_AA00.js
2
3   var info = require("./AA00info.js");
4
5   info("aa00", "Redwoods", "010-1234-1234");
6   info("aa55", "COMSI", "010-1234-5678");
7
```

문제 출력 디버그 콘솔 터미널

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt02\start>node index_aa00
My Info
ID : aa00
Name : Redwoods
Phone : 010-1234-1234

My Info
ID : aa55
Name : COMSI
Phone : 010-1234-5678
```

Save as
AAnn_info.png

wk02 : Practice-01 : AAnn_Rpt02

- [Target of this week]

  My Info using node module – aanninfo.js

Upload folder : aax-nn-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

[Upload to github]

- [wk02]
  - upload all work of this week
  - Use repo “aax-nn” in github
  - upload folder “aax-nn-rpt02” in your github.

Node Server II.

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

7.1.1 Express server test

Step 1 : npm init

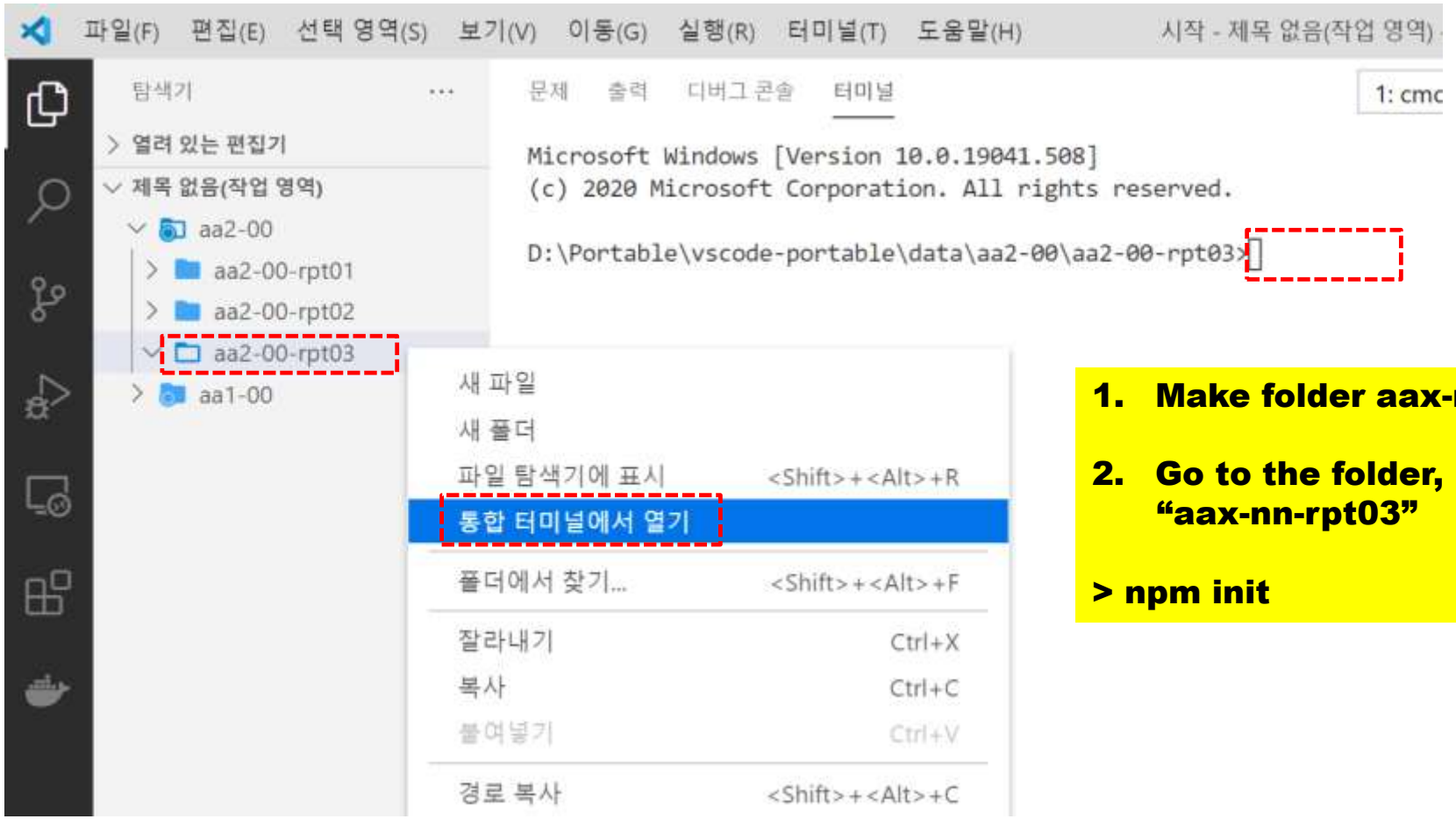
Step 2 : npm install --save express

Step 3 : Write Express code

Step 4 : Run app.js

Step 5 : http://localhost:3000

7.1.2 Express server test



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

> npm init

7.1.2 Express server test: npm init

문제 출력 디버그 콘솔 터미널 1: node

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-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: (aa2-00-rpt03) expresstest
version: (1.0.0)
description: test express server
entry point: (index.js) app.js
test command:
git repository:
keywords:
author: aa00
license: (ISC) MIT
About to write to D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt03\package.json:
```

7.1.3 Express server test: package.json

package.json

```
 D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt03 디렉터리

2020-09-15  오후 02:41    <DIR>          .
2020-09-15  오후 02:41    <DIR>          ..
2020-09-15  오후 02:41               228 package.json
               1개 파일                 228 바이트
               2개 디렉터리  2,515,891,486,720 바이트 남음
```

package.json ×

aa2-00 > aa2-00-rpt03 > package.json > ...

```
1   {
2     "name": "expresstest",
3     "version": "1.0.0",
4     "description": "test express server",
5     "main": "app.js",
      ▷ Debug
6     "scripts": {
7       "test": "echo \"Error: no test specified\" && exit 1"
8     },
9     "author": "aa00",
10    "license": "MIT"
11  }
```

7.1.4 Express server test: express module install

npm install –save express

```
문제   출력   디버그 콘솔   터미널                                    1: cmd

D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt03>npm install --save express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN expresstest@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 2.239s
found 0 vulnerabilities
```

```
D:\Portable\vscode-portable\data\aa2-00\aa2-00-rpt03 디렉터리

2020-09-15  오후 02:44    <DIR>          .
2020-09-15  오후 02:44    <DIR>          ..
2020-09-15  오후 02:44    <DIR>          node_modules
2020-09-15  오후 02:44            14,287 package-lock.json
2020-09-15  오후 02:44               278 package.json
               2개 파일              14,565 바이트
```

7.1.5 Express server test: express install

package.json

aa2-00 > aa2-00-rpt03 > package.json > ...

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

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

7.1.6 Express server test: app.js

package.json | app.js

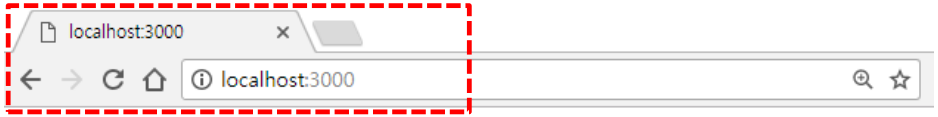
aa2-00 > aa2-00-rpt03 > app.js > ...

```
1  // app.js
2  var express = require("express");
3  var app = express();
4  var port = 3000;
5
6  app.get("/", function (req, res) {
7    res.send('<a href="/hello">Hello Page</a>');
8  });
9
10 app.get("/hello", function (req, res) {
11   res.send("Hello aa00");
12 });
13
14 app.get("/comsi", function (req, res) {
15   res.send("Hello Comsi!");
16 });
17
18 // listening requests from clients
19 var server = app.listen(port, function () {
20   console.log("Listening on port %d", server.address().port);
21 });
```

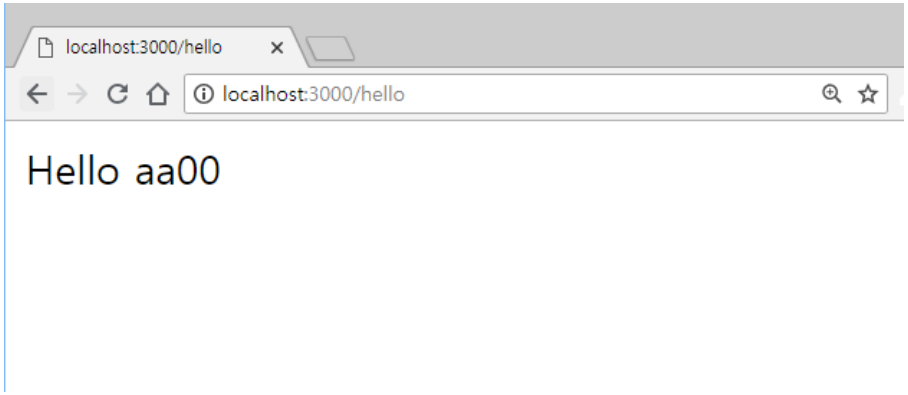
Express server

routing

7.1.7 Express server test: run app.js

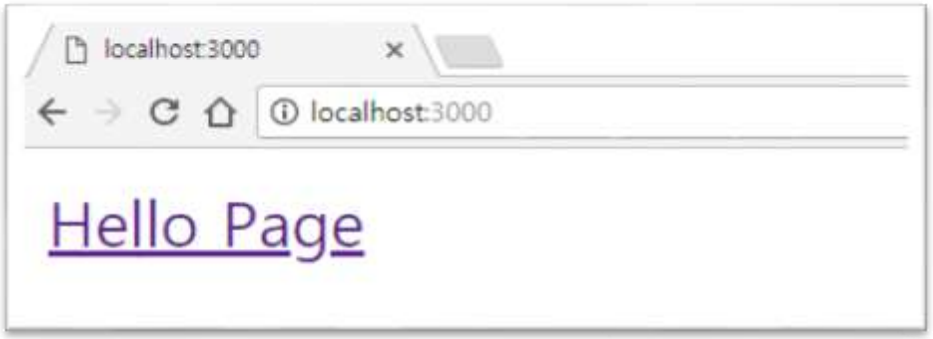


Hello Page

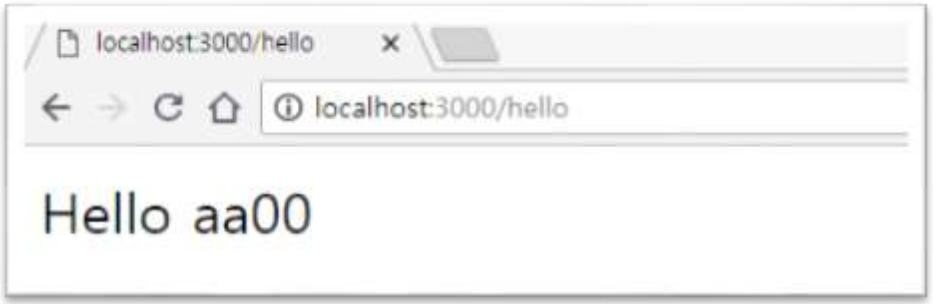


7.1.8 Express server test: test server routing

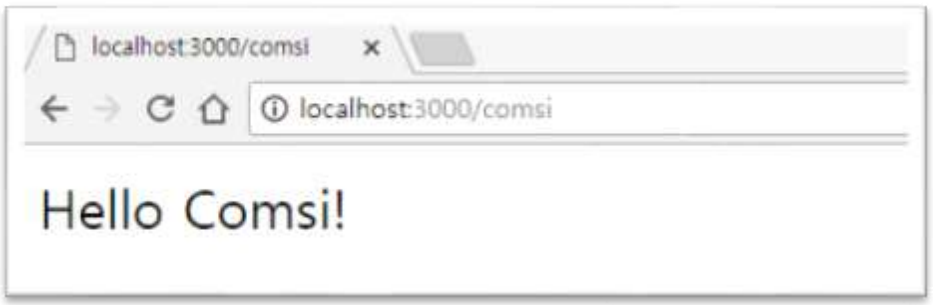
localhost:3000



localhost:3000/hello



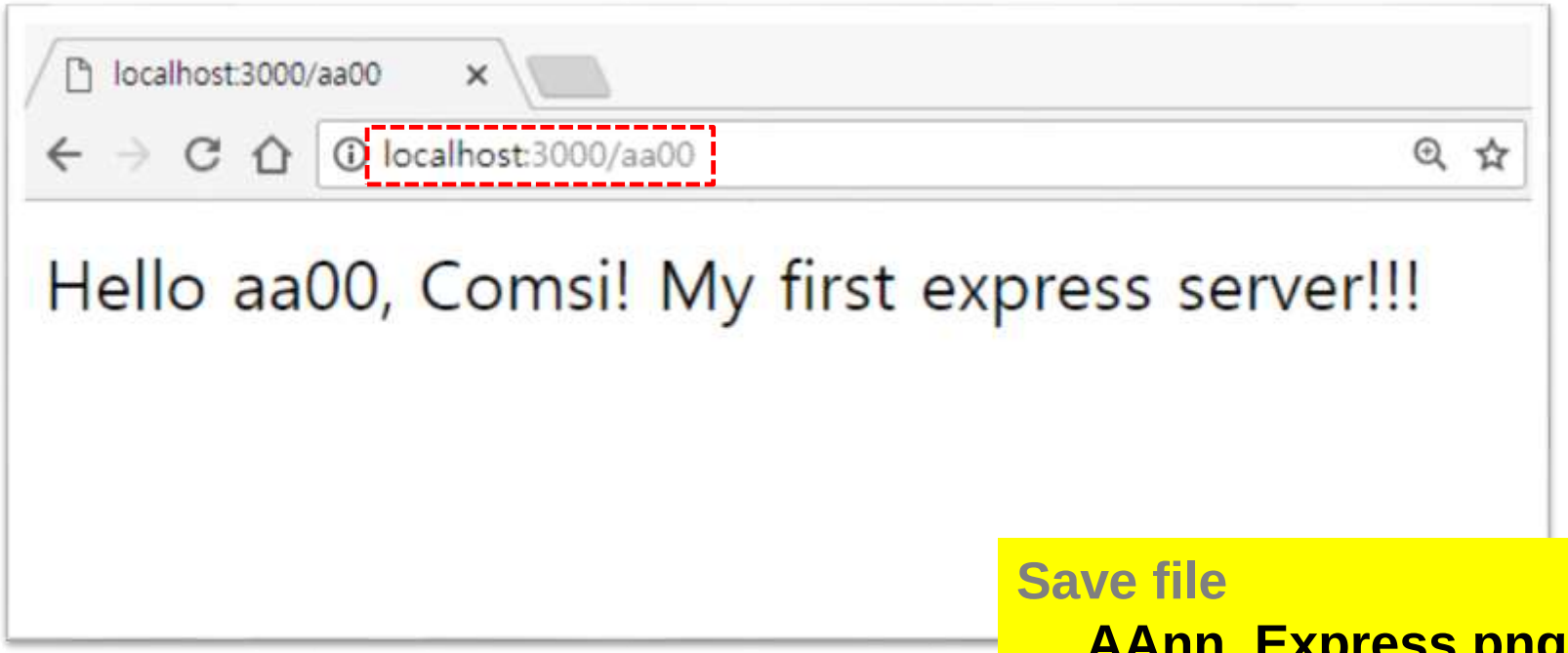
localhost:3000/comsi



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!!!');
});
```

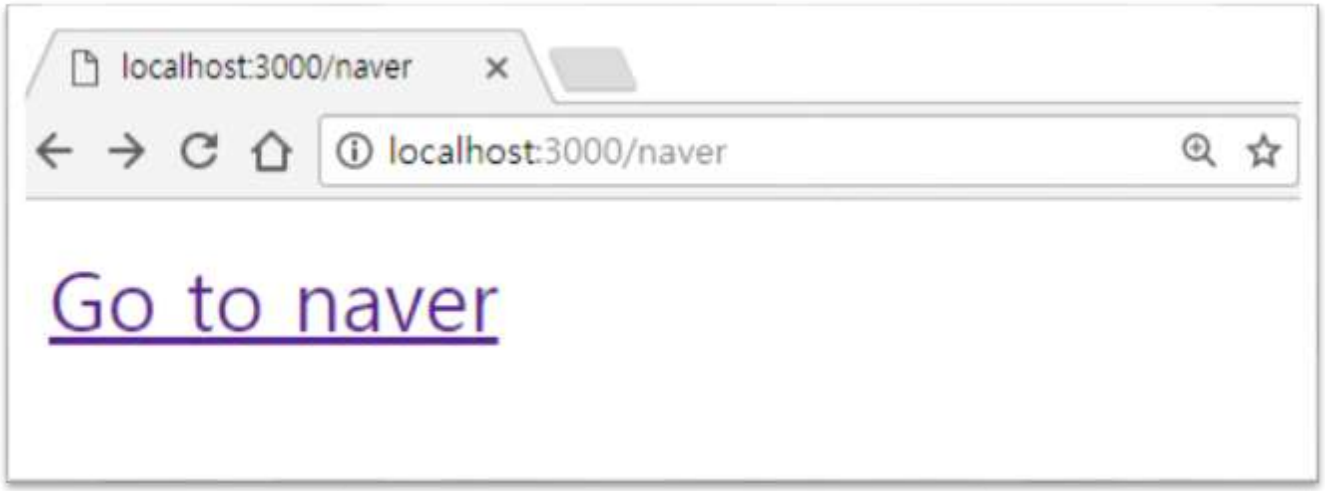


Save file
AAnn_Express.png

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



[Hint] **Go to naver**

[Practice]

- [wk03]
  - Express server
  - Upload folder: aax-nn-rpt03
  - Use repo “aax-nn” in github

wk03 : Practice-02 : AAnn_Rpt03

- [Target of this week]
  - Complete your works
  - Save your outcomes and 1 figure

Upload folder : aax-nn_rpt03

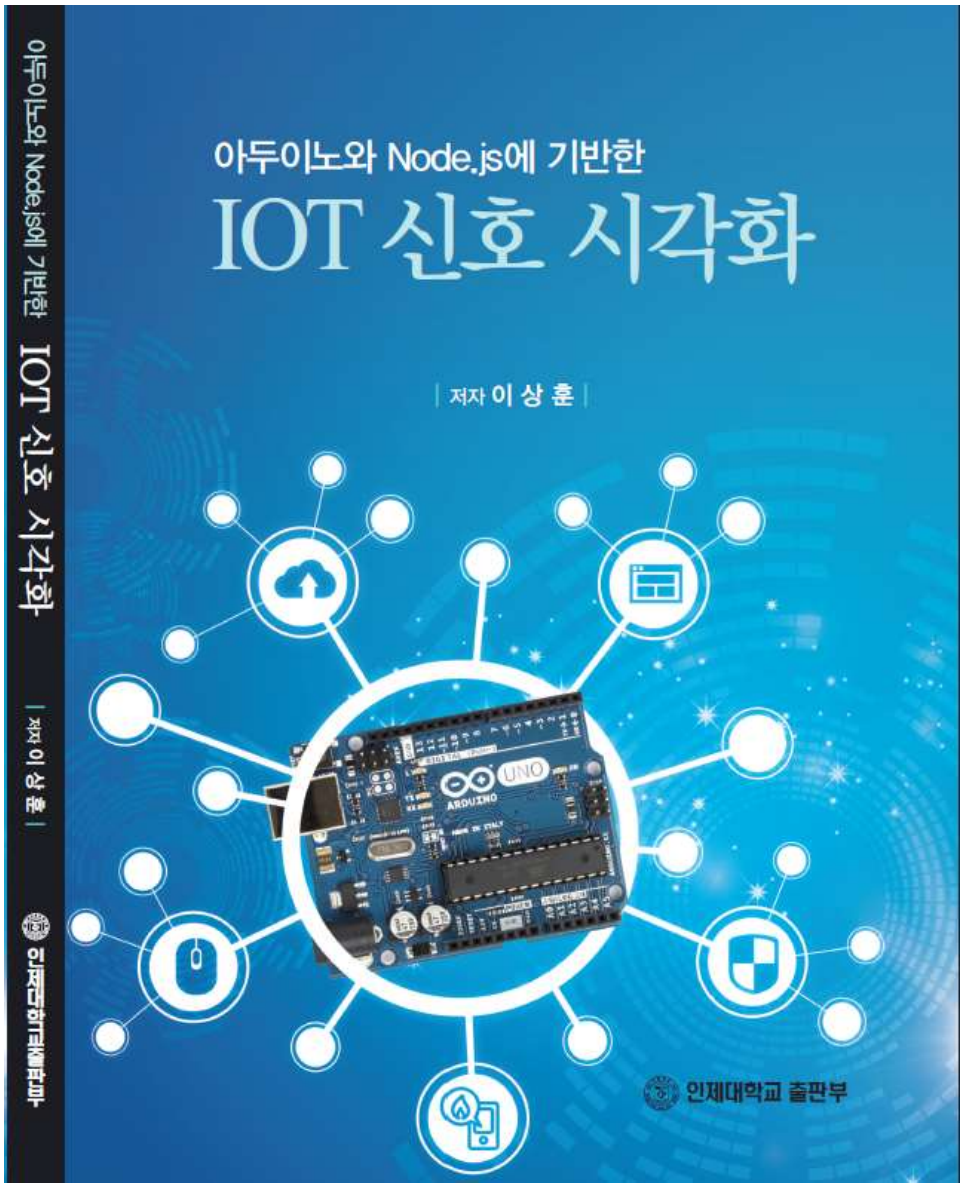
- 제출할 파일들
  - ① **AAnn_Express.png**
  - ② **app.js**

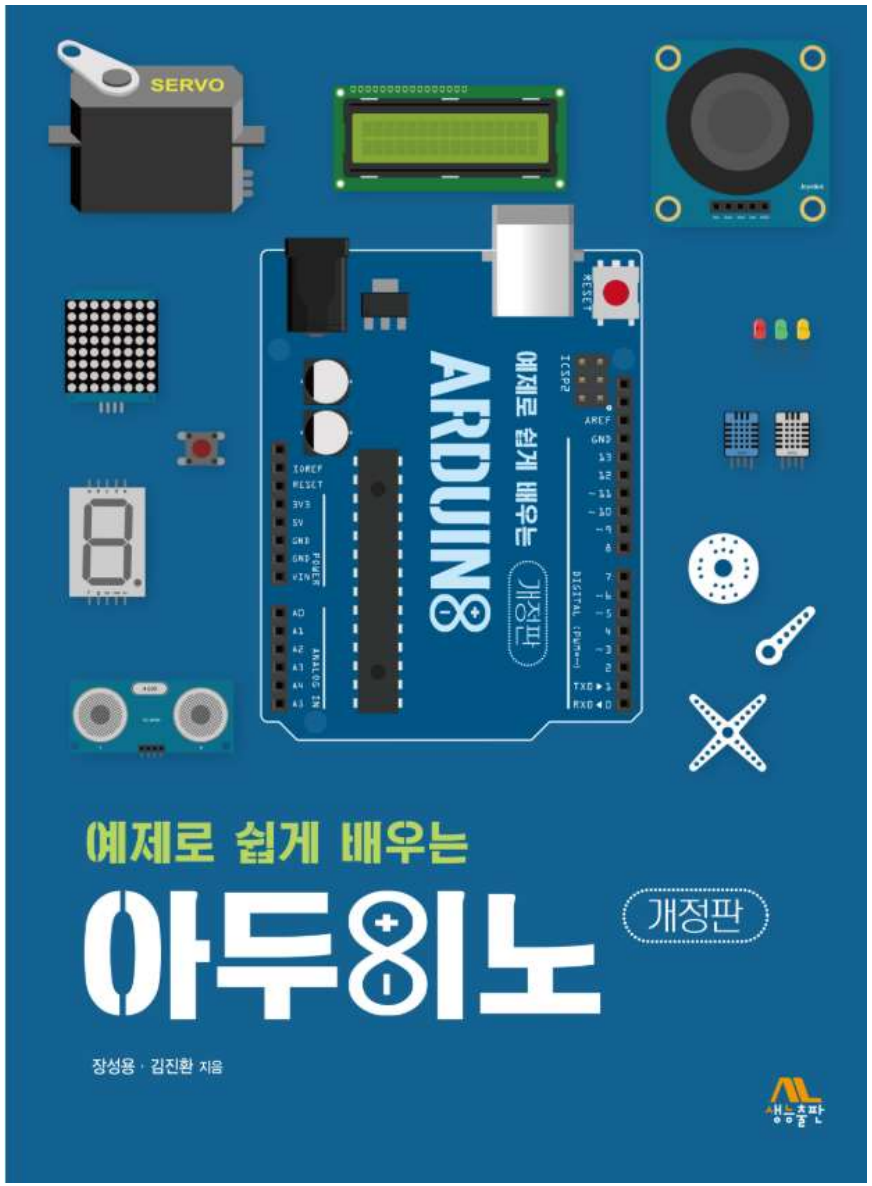
Lecture materials

References & good sites

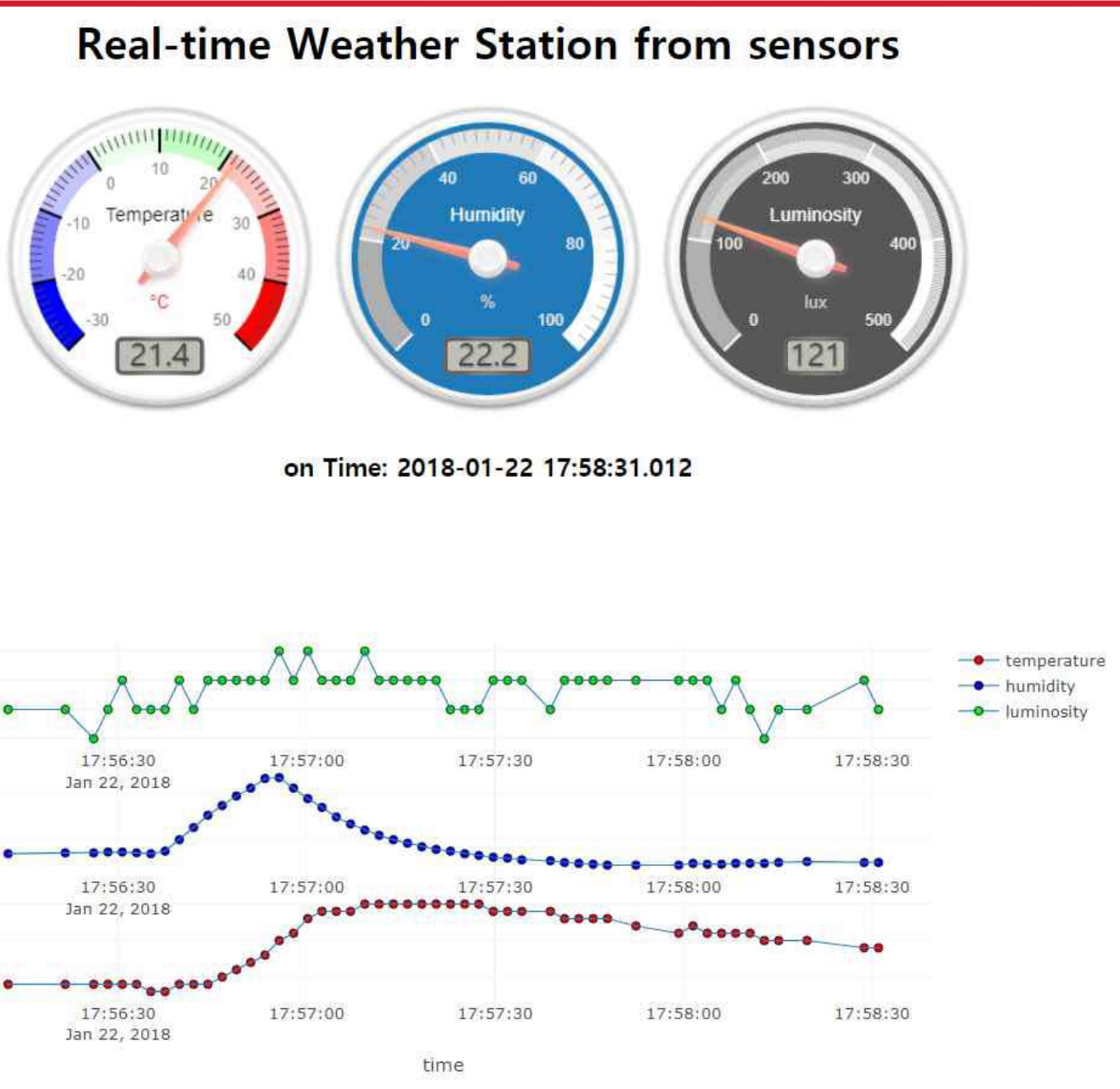
- ✓ http://www.arduino.cc Arduino Homepage
- ✓ http://www.nodejs.org/ko Node.js
- ✓ https://plot.ly/ plotly
- ✓ https://www.mongodb.com/ MongoDB
- ✓ http://www.w3schools.com By w3schools
- ✓ http://www.github.com GitHub

주교재 및 참고도서

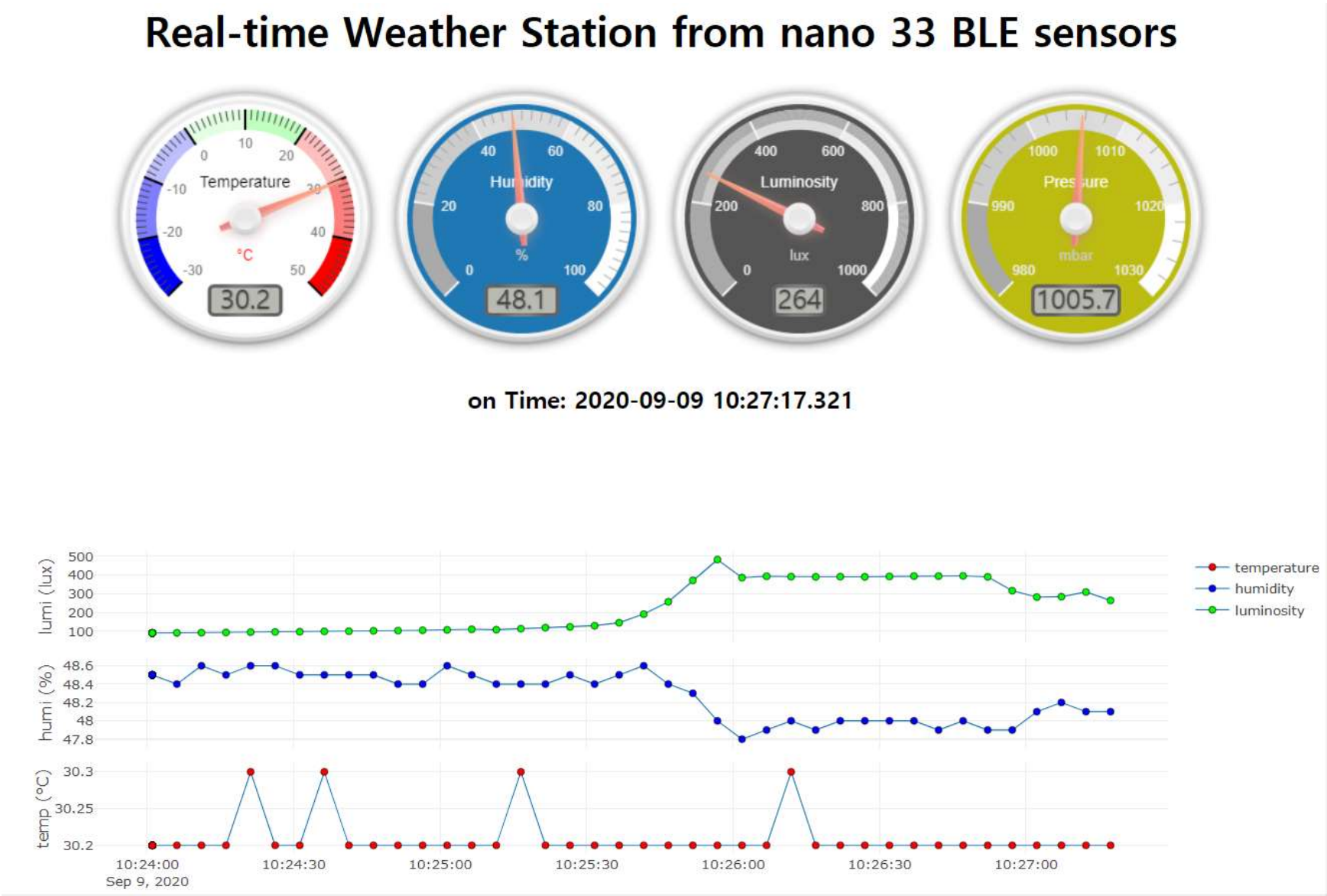

아두이노와 Node.js에 기반한
IOT 신호 시각화
| 저자 이 상 훈 |
인제대학교 출판부
아두이노와 Node.js에 기반한 IOT 신호 시각화
| 저자 이 상 훈 |
인제대학교 출판부


SERVO
ARDUINO
예제로 쉽게 배우는
개정판
예제로 쉽게 배우는
아두이노
개정판
장성용 · 김진환 지음

Target of this class



Target of this class



Another target of this class