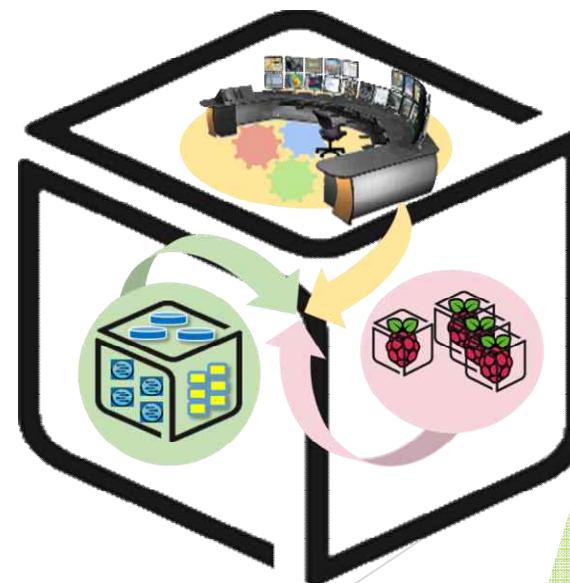


# SmartX Labs

## for computer systems

WebApp Lab  
version 1.0



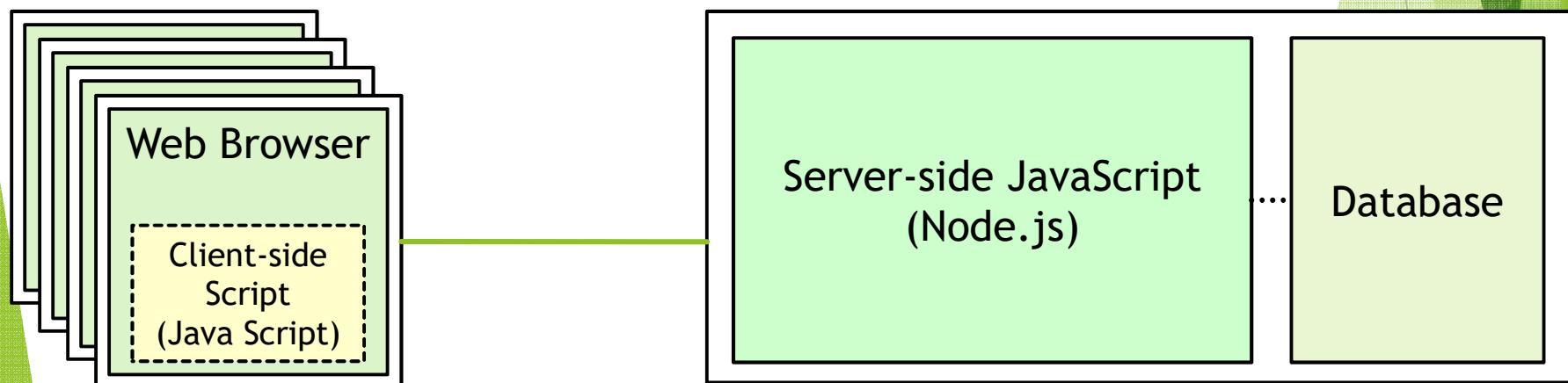
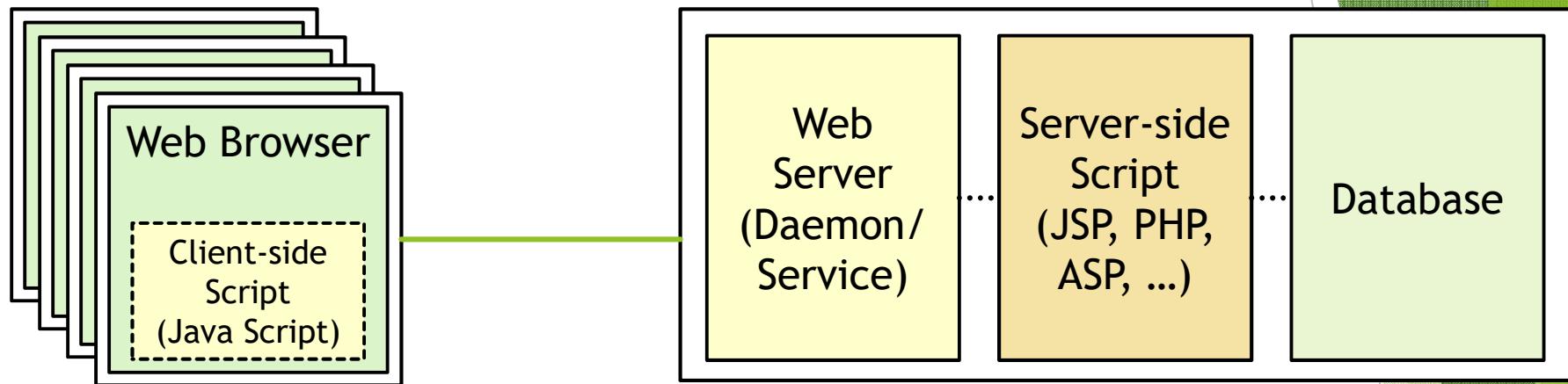
# History and Contributor of Box Lab

## (2016. 05. 22)

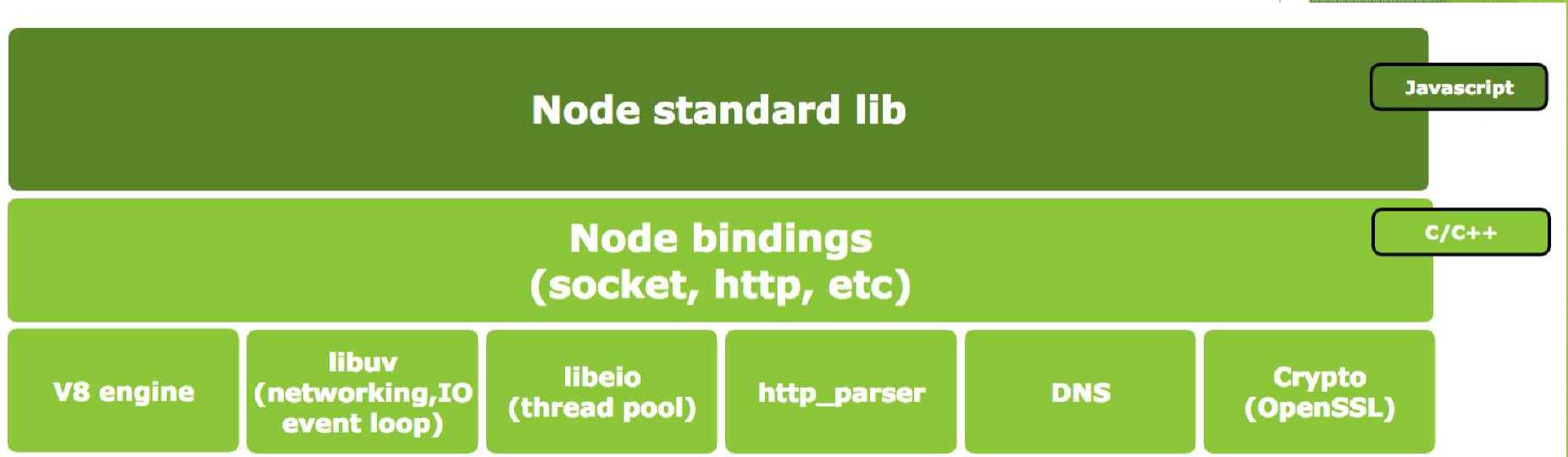
Version	Updated Date	Updated Contents	Contributor
			Lucas
V1.0	2016-05-22		이준기

# 웹 서버 프로그래밍

## ▶ Node.js 이전 (e.g LAMP)



# Node.js Architecture

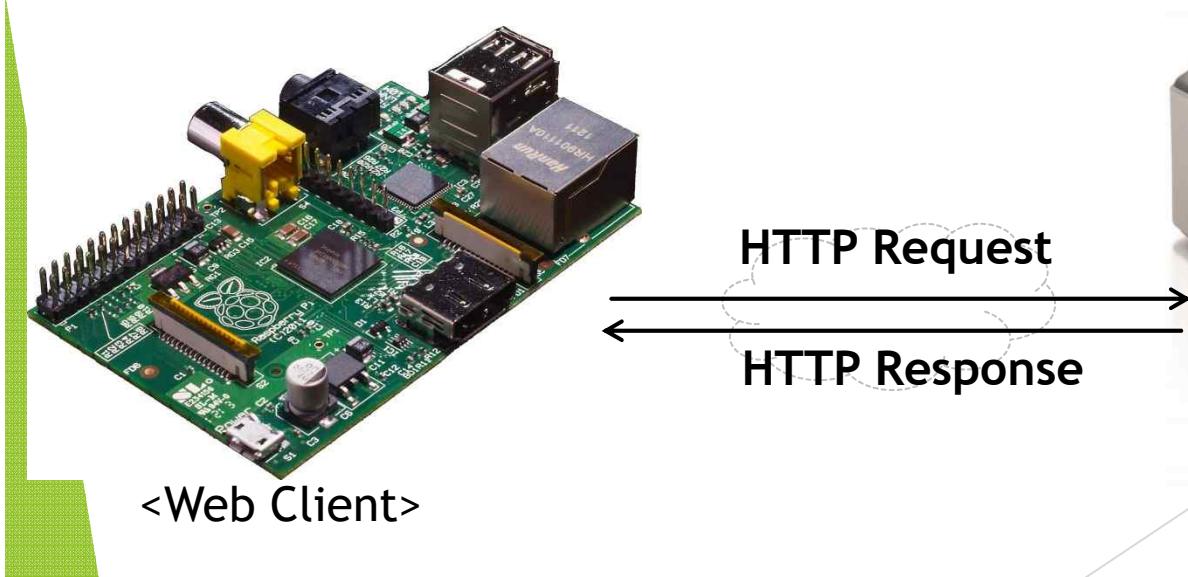


# Node.js Pros and Cons

- ▶ 장점
  - ▶ Server-side JavaScript → 높은 생산성
  - ▶ Single thread, non-block I/O → 가볍고 빠름
  - ▶ I/O 직접수행 안 함 → 프로세스 Block되지 않음
- ▶ 단점
  - ▶ Single thread → 멀티코어 CPU 효율을 위해 여러 개의 프로세스를 사용해야 하고, 어떤 한 작업이 무거우면 전체 성능이 저하될 수 있음
  - ▶ Event Callback → 중첩될 경우 가독성 급격히 저하 (callback hell)
  - ▶ V8 engine → Garbage Collection기반 메모리관리로 순간적인 CPU 사용률 상승 가능성 있고, 이는 서버 안정성 저하

# 개발환경 구축

- ▶ Node.js 설치 (Ubuntu 14.04)
  - ▶ \$ sudo apt-get update
  - ▶ \$ sudo apt-get install nodejs npm



OS: Ubuntu 14.04  
Web-Server: Node.js

# Hello World

Ex) vim hello.js

```
console.log('Hello World');
```

- ▶ How to run?

```
$ nodejs hello.js
```

```
lucas@Mesos-Ctrl:~/nodejs/hello$ nodejs hello.js
Hello World
lucas@Mesos-Ctrl:~/nodejs/hello$ █
```

# Nodeclipse “Enide 2015”

## : Node.js development on Ubuntu

- ▶ Install JAVA JDK 8
- ▶ \$ sudo add-apt-repository ppa:webupd8team/java
- ▶ \$ sudo apt-get update
- ▶ \$ sudo apt-get install oracle-java8-installer



# Nodeclipse “Enide 2015”

## : Node.js development on Ubuntu

- ▶ Install nodeclipse
- ▶ Download Nodeclipse “Enide 2015”
  - ▶ [https://sourceforge.net/projects/nodeclipse/?source=typ\\_redirect](https://sourceforge.net/projects/nodeclipse/?source=typ_redirect)
- ▶ Extract to folder
- ▶ \$ sudo nautilus ~/Downloads sudo nautilus /opt
- ▶ \$ cd chmod -R 7555 eclipse
- ▶ \$ sudo gedit /usr/share/applications/enide-2015-7.desktop



# Simple TCP Server

```
var net = require('net');

var server = net.createServer(function (socket) {
  socket.write('Echo server\r\n');
  socket.pipe(socket);
});

server.listen(1337);
```

```
lucas@Mesos-Ctrl:~$ telnet 127.0.0.1 1337
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
Echo server
|
```

```
root@Mesos-Ctrl:/home/lucas# curl telnet://127.0.0.1:1337
Echo server
^C
root@Mesos-Ctrl:/home/lucas# curl 127.0.0.1:1337
Echo server
GET / HTTP/1.1
User-Agent: curl/7.35.0
Host: 127.0.0.1:1337
Accept: */*
```

<Client>

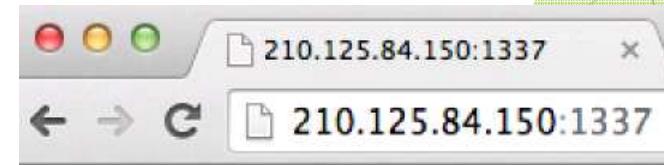
# Simple Web Server

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337);
console.log('Server running at port 1337/');
```

```
lucas@Mesos-Ctrl:~/nodejs/simple$ nodejs simple.js
Server running at port 1337/
```

<Server>

```
lucas@Mesos-Ctrl:~$ curl 127.0.0.1:1337
Hello World
lucas@Mesos-Ctrl:~$
```



Hello World

<Client>

# External File Execution (Simple)

```
var exec = require('child_process').exec;  
exec("date", function (error, stdout, stderr) {  
  console.log(stdout);  
});
```

```
lucas@Mesos-Ctrl:~/nodejs/external$ nodejs app.js  
Thu Nov 27 16:38:13 KST 2014  
  
lucas@Mesos-Ctrl:~/nodejs/external$ █
```

# External File Execution (Web)

```
var http = require('http'),  
  exec = require('child_process').exec;  
  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/plain'});  
  exec("date", function (error, stdout, stderr)  
  {res.end(stdout)} );  
}).listen(1337);  
console.log('Server running');
```

```
lucas@Mesos-Ctrl:~/nodejs$ curl 127.0.0.1:1337  
Thu Nov 27 16:34:27 KST 2014  
lucas@Mesos-Ctrl:~/nodejs$ curl 127.0.0.1:1337  
Thu Nov 27 16:34:32 KST 2014  
lucas@Mesos-Ctrl:~/nodejs$ curl 127.0.0.1:1337  
Thu Nov 27 16:34:34 KST 2014  
lucas@Mesos-Ctrl:~/nodejs$ curl 127.0.0.1:1337  
Thu Nov 27 16:34:35 KST 2014  
lucas@Mesos-Ctrl:~/nodejs$ █
```

# Node Packaged Module(NPM)

- ▶ Node.js로 만들어진 모듈 관리자
  - ▶ Ubuntu의 APT와 유사하다
- ▶ 패키지 설치
  - ▶ npm install <package name>
  - ▶ Global install: 패키지를 /usr/local/lib에 설치, 패키지에 따라 Global로 설치해야 하는 것이 있다.
  - ▶ Local install: 패키지를 현재 폴더 내에 설치하고, 폴더내의 파일에서만 불러올 수 있다.

# File Upload (1/4)

- ▶ 패키지 설치: npm install formidable  
fs-extra

```
var formidable = require('formidable'),
    http = require('http'),
    util = require('util'),
    fs   = require('fs-extra');

http.createServer(function(req, res) {
  /* Process the form uploads */
  if (req.url == '/upload' && req.method.toLowerCase() == 'post') {
    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}));
    });
  }
});
```

## File Upload (2/4)

```
form.on('progress', function(bytesReceived, bytesExpected) {  
    var percent_complete = (bytesReceived / bytesExpected) * 100;  
    console.log(percent_complete.toFixed(2));  
});  
  
form.on('error', function(err) {  
    console.error(err);  
});  
  
form.on('end', function(fields, files) {  
    /* Temporary location of our uploaded file */  
    var temp_path = this.openedFiles[0].path;  
    /* The file name of the uploaded file */  
    var file_name = this.openedFiles[0].name;  
    /* Location where we want to copy the uploaded file */  
    var new_location = process.env.PWD + '/';
```

# File Upload (3/4)

```
fs.copy(temp_path, new_location + file_name, function(err) {  
    if (err) {  
        console.error(err);  
    } else {  
        console.log("success!")  
    }  
});  
});  
  
return;  
}  
  
/* Display the file upload form. */  
res.writeHead(200, {'content-type': 'text/html'});  
res.end()
```

# File Upload (4/4)

```
'<form action="/upload" enctype="multipart/form-data" method="post">'+  
'<input type="text" name="title"><br>'+  
'<input type="file" name="upload" multiple="multiple"><br>'+  
'<input type="submit" value="Upload">'+  
'</form>'  
);  
  
}).listen(1337);
```

```
lucas@Mesos-Ctrl:~/nodejs/fupload$ nodejs upload.js  
1.02  
2.03  
3.05  
4.06  
5.08  
6.10 96.52  
7.11 97.54  
8.13 98.56  
9.14 99.57  
10.16 100.00  
11.18 success!  
12.19 ^C  
lucas@Mesos-Ctrl:~/nodejs/fupload$ ls  
kernel.img  node_modules  upload.js  
lucas@Mesos-Ctrl:~/nodejs/fupload$
```

```
lucas@Mesos-Ctrl:~$ curl -F upload=@kernel.img http://127.0.0.1:1337/upload  
received upload:  
  
{ fields: {},  
  files:  
    { upload:  
      { domain: null,  
       _events: {},  
       _maxListeners: 10,  
       size: 6449964,  
       path: '/tmp/8374da21ed85f5d42f45325d112f65fd',  
       name: 'kernel.img',  
       type: 'application/octet-stream',  
       hash: null,  
       lastModifiedDate: Thu Nov 27 2014 21:05:37 GMT+0900 (KST),  
       _writeStream: [Object] } } }lucas@Mesos-Ctrl:~$  
lucas@Mesos-Ctrl:~$
```

# Express Framework

- ▶ Node.js web application framework
- ▶ 설치: npm install express
- ▶ 설치된 패키지 목록: npm ls or npm -g ls

# Web Server using Express

```
var express = require('express')
, http = require('http')
, app = express()
, server = http.createServer(app);
app.get('/', function (req, res) {
  res.send('Hello /\n');
});
app.get('/world.html', function (req, res) {
  res.send('Hello World\n');
});
server.listen(8888, function() {
  console.log('Express server listening on port ' + server.address().port);
});
```

```
lucas@Mesos-Ctrl:~/nodejs/express$ nodejs express.js
Express server listening on port 8888
```

```
lucas@Mesos-Ctrl:~/nodejs/express$ curl 127.0.0.1:8888
Hello /
lucas@Mesos-Ctrl:~/nodejs/express$ curl 127.0.0.1:8888/world.html
Hello World
lucas@Mesos-Ctrl:~/nodejs/express$
```

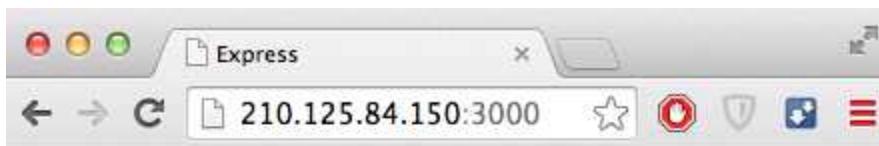
# Template Engine: Embedded JavaScript (for Dynamic Web Page)

- ▶ 패키지 설치 (전역)
  - ▶ `sudo npm install -g express-generator`
- ▶ 환경변수 설정
  - ▶ `export NODE_PATH=/usr/local/lib/node_modules`
- ▶ 작업폴더 생성 및 이동
  - ▶ `mkdir ejs`
  - ▶ `cd ejs`
- ▶ 스크립트 수정
  - ▶ `sudo vim /usr/local/lib/node_modules/express-generator/bin/express`
  - ▶ 첫행의 `node` → `nodejs`로 수정 후 저장
- ▶ 관련 패키지 환경설정
  - ▶ `/usr/local/lib/node_modules/express-generator/bin/express -e 실행`
- ▶ 의존 패키지 설치
  - ▶ `npm install`

# Template Engine: Embedded JavaScript (for Dynamic Web Page)

- ▶ 템플릿 설치 확인과 웹서버 앱 실행

```
lucas@Mesos-Ctrl:~/nodejs/ejs$ ls
app.js  bin  node_modules  package.json  public  routes  views
lucas@Mesos-Ctrl:~/nodejs/ejs$ nodejs bin/www
GET / 200 12.717 ms - 207
GET /stylesheets/style.css 200 4.195 ms - 110
GET / 304 2.075 ms - -
GET /stylesheets/style.css 304 0.736 ms - -
GET / 304 0.752 ms - -
GET /stylesheets/style.css 304 0.426 ms - -
GET / 304 1.112 ms - -
GET /stylesheets/style.css 304 0.963 ms - -
```



**Express**

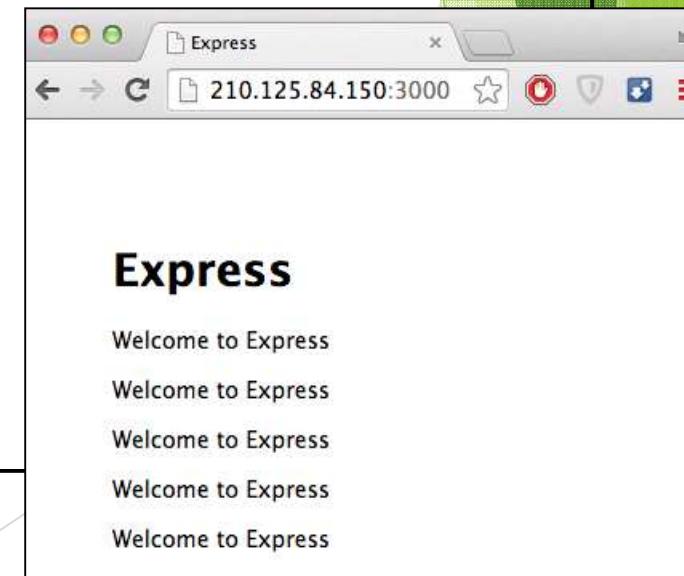
Welcome to Express

# Template Engine: Embedded JavaScript (for Dynamic Web Page)

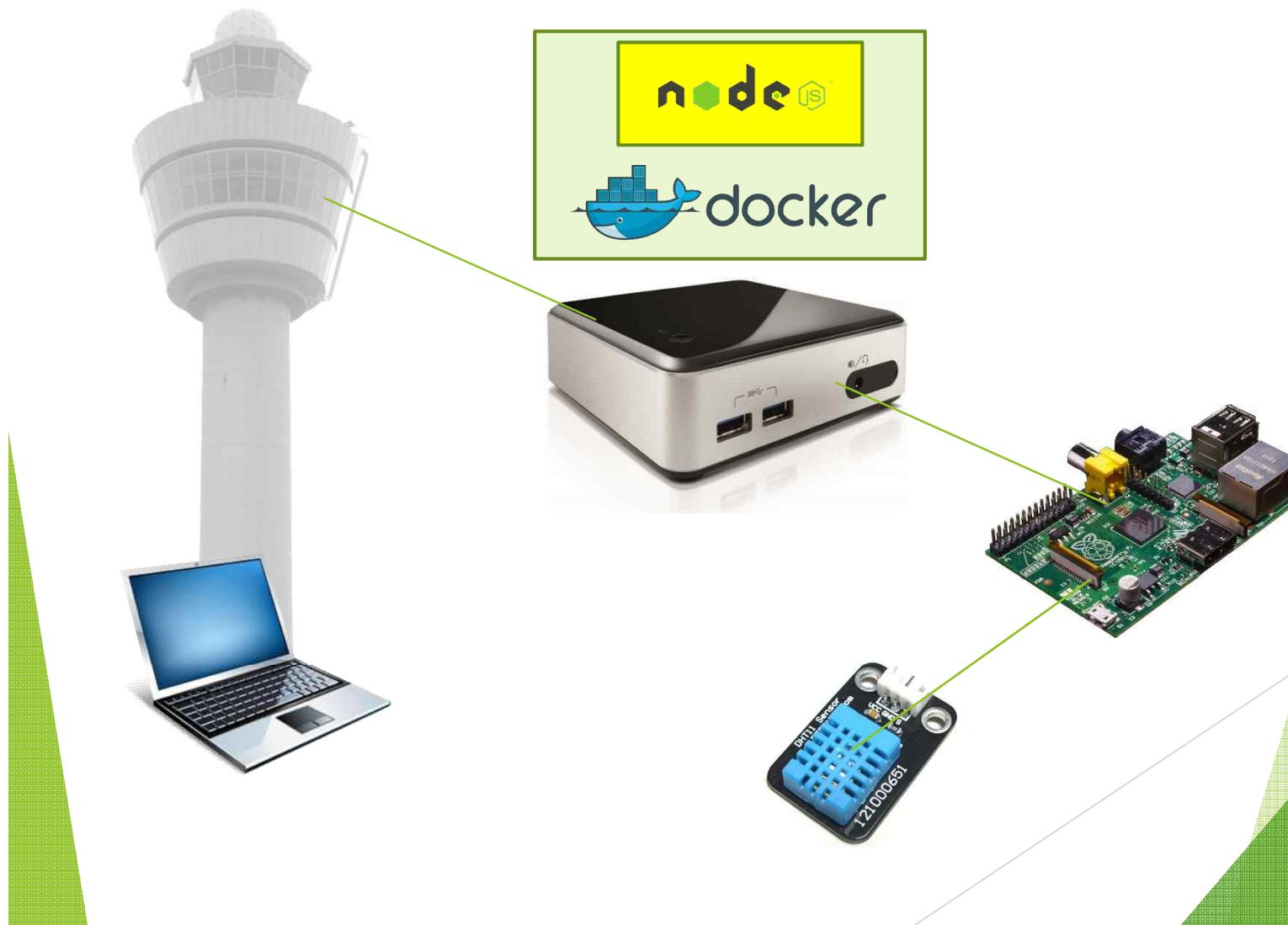
- ▶ 텍스트 반복출력 실습 (코드 수정)
  - ▶ vim views/index.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    <h1><%= title %></h1>
    <% for (var i = 0; i < 5; i++) { %>
      <p>Welcome to <%= title %></p>
    <% } %>
  </body>
</html>
```

- ▶ nodejs bin/www



# Architecture of WebApp



# Create Docker Container for nodejs WebApp

- ▶ \$ sudo docker run -it --net=host --name=webapp ubuntu /bin/bash
- ▶ # apt-get update
- ▶ # apt-get install net-tools
- ▶ # apt-get install iputils-ping

```
root@5daf51f2abb0:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:05
          inet  addr:172.17.0.5  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:5/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                  RX packets:4635 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:3305 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:12293854 (12.2 MB)  TX bytes:223481 (223.4 KB)

lo       Link encap:Local Loopback
          inet  addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING  MTU:65536  Metric:1
                  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```



# Install node.js in Docker Container

- ▶ # apt-get install nodejs
- ▶ (# apt-get install nano)

- ▶ Test
- ▶ # nano test.js

```
GNU nano 2.5.3
console.log('Hello World');
```

- ▶ # nodejs test.js

```
root@netcsnuc:/# nodejs test.js
Hello World
```



# nodejs WebApp code

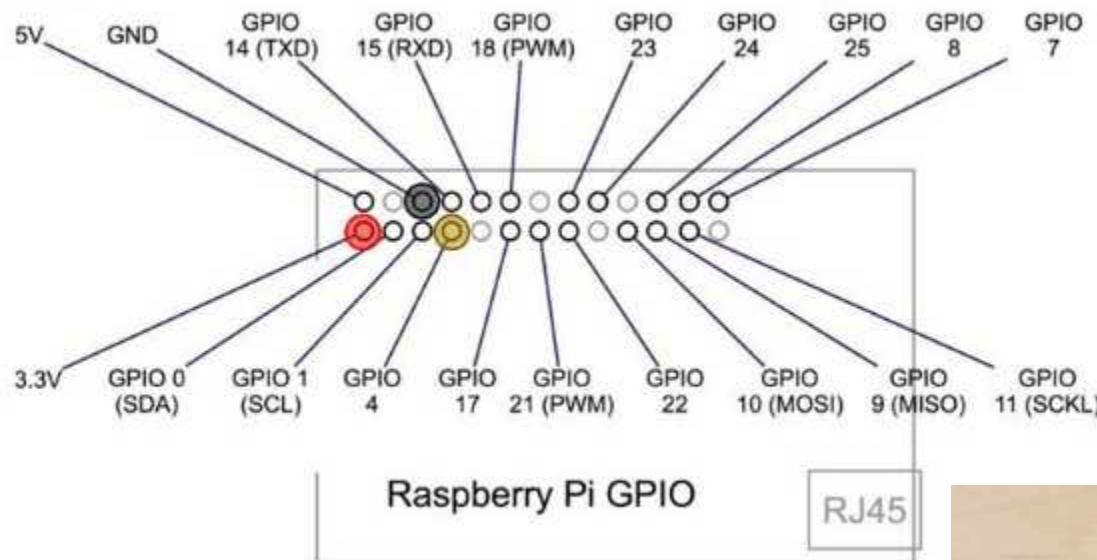
## ► # nano webapp.js

```
1 var http = require('http');
2 var url  = require('url');
3 var fs   = require('fs');
4 var temp="";
5
6 http.createServer(function (request, response) {
7
8     var query = url.parse(request.url, true).query
9
10    response.writeHead(200, { 'Content-Type': 'text/html' });
11    console.log(JSON.stringify(query));
12
13
14    if(JSON.stringify(query).length>13)
15    {
16        fs.writeFile('temp.txt', JSON.stringify(query), 'utf8', function (error){
17            console.log('write');
18        });
19    }
20    fs.readFile('temp.txt', 'utf8', function (error, data) {
21        console.log(data);
22        temp = data;
23    });
24    response.end(temp);
25
26 }).listen(80, function () {
27     console.log('Server running...');
28});
```

# nodejs WebApp code

- ▶ # nodejs webapp.js
- ▶ At the NUC, Open browser and goto
  - ▶ [http://<IP\\_of\\_NUC>](http://<IP_of_NUC>)

# WebApp with RPi sensor



<http://www.uugear.com/portfolio/dht11-humidity-temperature-sensor-module/>



# WebApp with RPi sensor

- ▶ Install dependencies at RPi
  - ▶ \$ sudo apt-get update
  - ▶ \$ sudo apt-get install python-pip
  - ▶ \$ sudo apt-get install libpython2.7-dev python-numpy
  - ▶ \$ sudo pip install RPi.GPIO
  - ▶ \$ sudo apt-get install mercurial



# WebApp with RPi sensor

- ▶ Temperature sensor code
  - ▶ \$ sudo nano RPI\_temp.py
  - ▶ Copy code from  
[https://github.com/2jungi/SmartX-Mini/blob/master/RPI\\_temp.py](https://github.com/2jungi/SmartX-Mini/blob/master/RPI_temp.py)



# Revise code for WebApp

- ▶ Change IP address to <Your NUC's IP>

```
104 | if int(Humidity) + int(Temperature) - int(bin2dec(crc)) == 0:  
105 |     print "Humidity:"+ Humidity +"%"  
106 |     print "Temperature:"+ Temperature +"C"  
107 |     urllib2.urlopen("http://192.168.88.85?temp="+Temperature+"humid="+Humidity).close
```

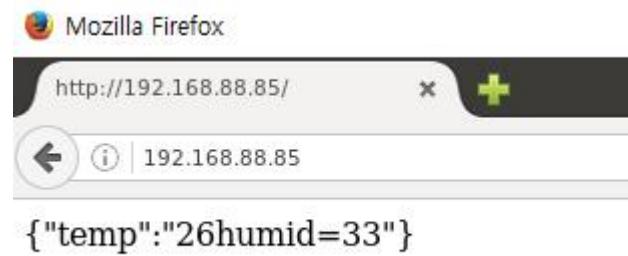
- ▶ sudo python RPI\_temp.py

```
HypriotOS: root@pi17 ~  
$ sudo python sample.py  
Humidity:30%  
Temperature:32C
```



# WebApp in Browser

- ▶ At the Docker container in NUC
  - ▶ # nodejs webapp.js
- ▶ At the NUC or DevTower
  - ▶ Open Web browser and goto `http://<IP_of_NUC>`
- ▶ At the RPi
  - ▶ \$ sudo python RPI\_temp.py



# References

- ▶ <http://nodejs.org/>
- ▶ <http://expressjs.com>
- ▶ <http://pyrasis.com/nodejs/nodejs-HOWTO>
- ▶ <http://www.codediesel.com/nodejs/processing-file-uploads-in-node-js/>
- ▶ <http://www.gliffy.com/publish/2752090/>
- ▶ <http://kipid.tistory.com/entry/Learning-Nodejs>
- ▶ <http://www.nodeclipse.org/ubuntu/linux/java/nodejs/2015/2015/07/09/Starting-with-Java-and-Node.js-development-on-Ubuntu-Linux.html>