

Node.js를 사용한 블로그 만들기

컴퓨터네트워크 6주차

공과대학 5호관 633호
데이터 네트워크 연구실

조교 : 황동준

실습

1. Node.js 웹 프레임워크 Express 예제

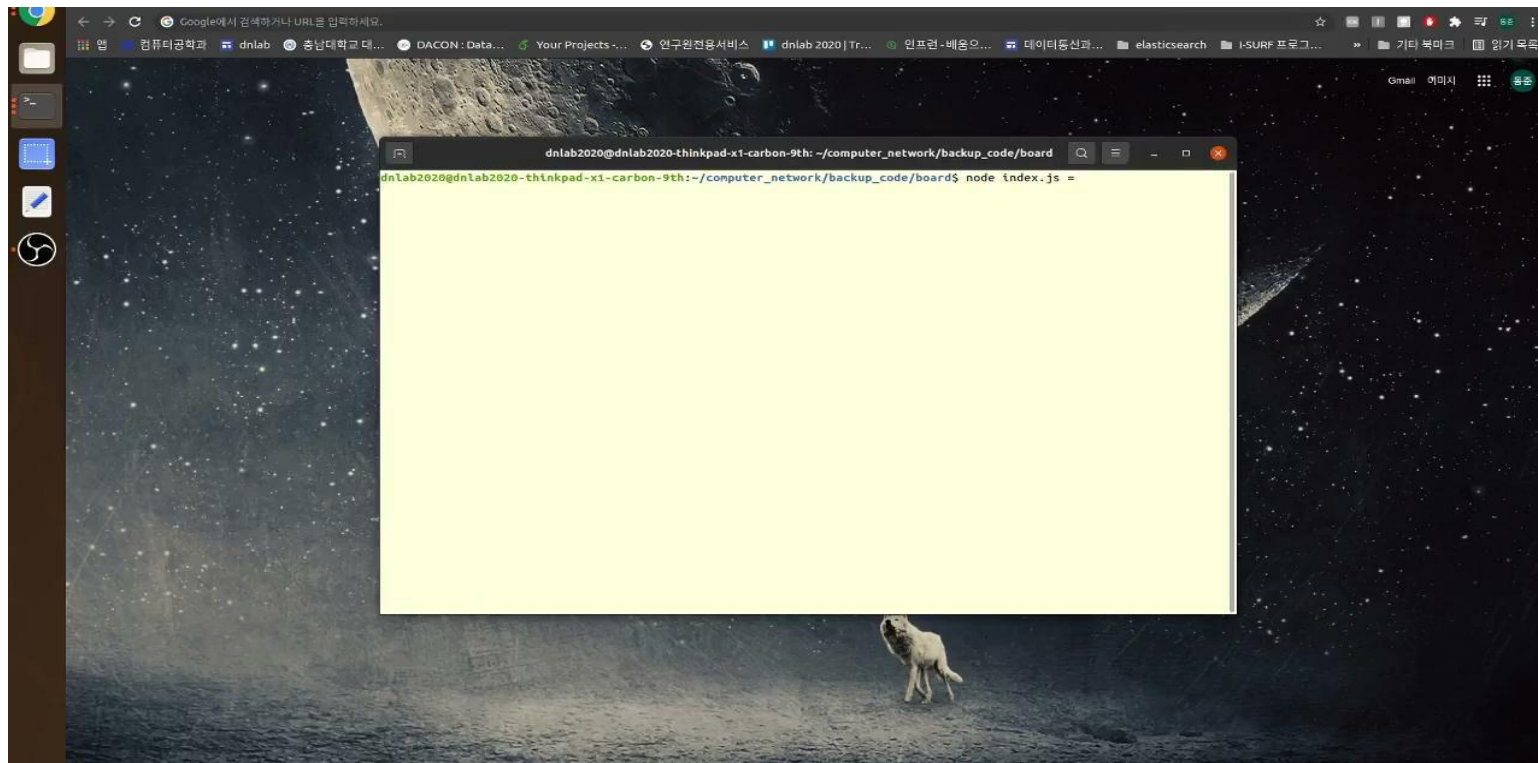
- Node.js 시작하기
- Express 설치 및 간단 사용법

과제

1. Node.js를 통한 블로그 생성 (10점)

- main (index.js) (4점)
- routes (home.js) (4점)
- 최종 결과 (2점)
 - 생성한 블로그 접속
 - 자신의 학번과 이름 확인
 - 결과 영상과 동일하게 진행!

과제 결과 영상



실습 : Node.js 웹 프레임워크 Express 예제

Node.js



- Node.js란?
 - Node.js는 Chrome V8 Javascript 엔진으로 빌드된 Javascript 런타임
 - Node.js의 패키지 생태계는 npm (e.g. python3의 pip3)
 - <https://perfectacle.github.io/2017/06/18/what-is-node-js/>
- Ubuntu 환경에 Node.js 설치하기
 - <https://soojae.tistory.com/25>
- 웹 프레임워크 Express 이해하기
 - <https://velopert.com/294>
- 공식 사이트
 - <https://nodejs.org/ko/>
- 관련 자료 사이트
 - <https://www.simform.com/blog/what-is-node-js/>

Node.js 시작하기 - 1

- “npm init”
 - node 프로그램을 시작(initialize)하는 명령어로 package.json 파일을 생성을 도움
 - package.json 파일은 해당 node 프로그램에 대한 기본 정보를 담고 있는 파일
- 오른쪽 파란색 사각형이 package.json 파일의 내용

```
datanetwork@datanetwork-VirtualBox:~/test_nodejs$ 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: (test_nodejs)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/datanetwork/test_nodejs/package.json:

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

Is this OK? (yes)
```

Node.js 시작하기 - 2

- “npm install express --save”
 - npm을 사용하여 express를 설치
 - --save옵션 : dependencies에 적용
- package.json파일에 dependencies의 내용이 추가가 된 것을 확인할 수 있음

```
datanetwork@datanetwork-VirtualBox:~/test_nodejs$ npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN test_nodejs@1.0.0 No description
npm WARN test_nodejs@1.0.0 No repository field.
```

```
+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 2.547s
found 0 vulnerabilities
```

New patch version of npm available! 6.14.4 → 6.14.8
Changelog: <https://github.com/npm/cli/releases/tag/v6.14.8>
Run `npm install -g npm` to update!

```
datanetwork@datanetwork-VirtualBox:~/test_nodejs$ ls
index.js  node_modules  package.json  package-lock.json
```

```
1 {
2   "name": "test_nodejs",
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  "dependencies": {
12    "express": "^4.17.1"
13  }
14 }
```

Express

```
1 const express = require('express');
2 const app = express();
3 const port = 3000;
4
5 app.get('/', (req, res) => {
6   res.send('Hello World!');
7 });
8
9 app.listen(port, () => {
10   console.log(`Example app listening at http://localhost:${port}`);
11 });
```

index.js

=

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

- express를 모듈을 불러와 객체 생성
 - require를 통해 모듈을 불러옴
- app.get → Get 방식으로 불러오겠다는 의미
 - (req, res) ⇒ Callback 함수 ⇒ <https://dalkomit.tistory.com/65>
- “node index.js” ⇒ 실행 명령어

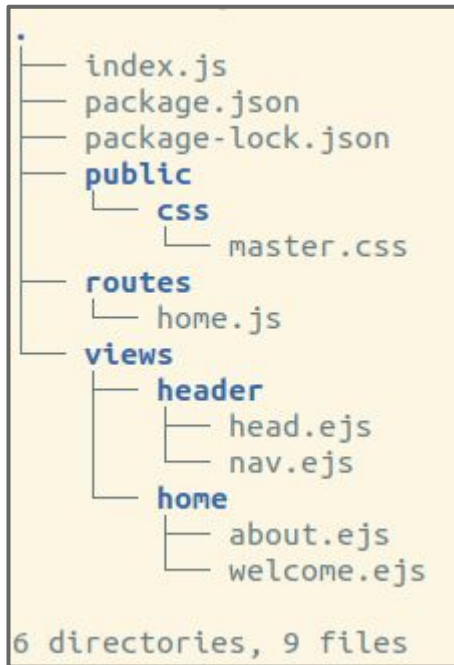
```
datanetwork@datanetwork-VirtualBox:~/test_nodejs$ node index.js
Example app listening at http://localhost:3000
```


**과제 : Node.js를 통한 블로그 생성
(Home, About)**

파일 트리 구조

- index.js
(main)
- public → css → master.css
(css)
- routes → home.js
(Router기능을 통해 페이지 경로 접속에 따른 render)
- views
 - header : Home에 대한 전체 페이지 ejs
 - home : Home에서 메뉴 클릭을 통해 이동하는 페이지 ejs

[참고] node_modules 디렉토리는 내부 파일이 너무 많아 잠시 이동시켜
tree구조 생성 (원래는 있어야함)



Main

```
1 var express = require('express');
2 var bodyParser = require('body-parser');
3 var methodOverride = require('method-override');
4 var app = express();
5
6 app.set('view engine', 'ejs');
7 app.use(express.static(__dirname+'/public'));
8 app.use(bodyParser.json());
9 app.use(bodyParser.urlencoded({extended:true}));
10 app.use(methodOverride('_method'));
```

} 모듈 호출 및 객체 생성

→ ejs파일 생성하여 HTML 문법 작성

→ 중첩된 객체 표현 허용

→ PUT, DELETE method 허용

- 모듈을 설치하여 호출 및 express 객체를 생성
- ejs는 express의 HTML 템플릿 (“npm install ejs --save”로 설치)
 - 파일이름.ejs로 생성하여 내용 작성 → view engine으로 인해 view디렉토리 경로 자동 인식

Routes

```
1 var express = require('express');  
2 var router = express.Router();
```

- express의 Router객체 생성
 - router.get() 함수를 통해 경로를 지정하여 render()함수 선언 가능
 - render() 함수는 해당 경로를 설정하여 경로의 html 페이지으로 렌더링

views / public

- 이번 views의 ejs파일들과 css파일은 제공
 - 참고할 점은 node.js파일과 ejs / css파일의 연관성을 살펴볼 것!
 - 추가적으로 결과 영상에서 보셨듯이 202050247 DongjunHwang으로 설정된 글을 **자신의 학번과 이름으로 수정**해서 보고서에 작성해야 함
- views
 - https://drive.google.com/file/d/1OsUZuhyNLeNBw20X_qusdFRXrQCrK6jv/view?usp=sharing
- public
 - <https://drive.google.com/file/d/1a8vrUWh-9bffTmD4c7RwyhZJ8Nx8Mdaj/view?usp=sharing>

과제 제출

- 과제 제출 기한
 - 2021년 10월 21일 17:59 까지 사이버 캠퍼스에 제출
- 제출 파일 (.zip)
 - npm init을 통해 지정한 파일 압축 (node_modules 디렉토리는 압축할필요 없음)
 - 보고서 (.pdf)
- 딜레이 1일당 20%감점
 - ex) 10점 만점 기준, 1일 딜레이인 경우 8점
- 과제 카피 적발시 보여준사람 본사람 모두 0점처리
 - 카피 적발기준 : 과제 유사도 80%이상

유의사항

- 파일명 : **CN02_06_학번_이름.zip**
 - npm init을 통해 지정한 파일을 압축하여 제출
 - 보고서 : **PDF로 제출할것 (파일명 : 06_학번_이름.pdf)**
 - 과제 목표 (도출해야할 결과)
 - 코드 설명과 과제 해결 방법 (실행 결과 사진도 보고서에 첨부, 코드 및 설명 필요)
 - 해당 부분 반드시 자세하게 작성할 것
 - 과제 느낀점 및 하고싶은 말 (**선택사항**)
 - 형식 지켜지지 않을시 채점대상에서 제외 (보고서도 HWP, DOC은 채점대상에서 제외)
- 질문
 - 디스코드 컴퓨터네트워크 채널에서 질문을 받음

부록

Node.js에 대해 알아보자



- Node.js 뒤에 붙은 접미사를 보면 .js인데, 자바스크립트와 관련된 것
 - 하지만 jQuery 처럼 클라이언트 사이트의 자바스크립트 라이브러리는 않음
 - 자바스크립트는 브라우저에서 구동되며, 웹 사이트에서 다양한 클라이언트단에서 프로그래밍적인 요소가 필요한 상황에서 사용되는 클라이언트 사이트 언어 ("[https://ko.wikipedia.org/wiki/클라이언트 사이트 스크립트 언어](https://ko.wikipedia.org/wiki/클라이언트_사이트_스크립트_언어)")
- node.js가 만들어진 주 목적은 서버측에서 구동되는 자바스크립트 개발 환경을 위해서임 (서버 사이트 스크립트 언어)
⇒ 서버에서 구동되어 동적인 콘텐츠를 만들어 내어 클라이언트에게 전달하는 역할
- node.js 를 요약해서 이야기 하자면 웹 서버 혹은 웹 어플리케이션 서버이면서 동시에 서버 사이트 스크립트의 역할도 하는 자바스크립트 기반의 서버 개발 환경
- node.js는 server를 다룰 수 있으며, javascript는 client(BOM, DOM)을 다룸
 - 반대로 node.js는 client를 다룰 수 없으며, javascript는 server를 다룰 수 없음

Node.js에 대해 알아보자



- 비동기 프로그래밍의 이해!
 - 비동기 프로그래밍의 기본은 비동기 API를 사용하여 비동기 이벤트를 발생시키고 완료여부를 알 수 있도록 Callback을 정의하는 것 (Node.js의 callback 함수)
- fs.readdir() 함수를 통해 이벤트가 발생
- 이벤트 완료되면 (err, filenames)에 값이 들어감
- err이나 filenames를 통해 추가적인 코드 구현 (“// callback”)

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

// non-blocking code
fs.readdir('.', function (err, filenames){
    // callback
});

console.log('can process next job...');
```

Node.js에 대해 알아보자

- Node.js와 V8의 관계
 - V8은 Javascript를 구문 분석하고 실행하는 node.js의 Javascript 엔진입
 - ex) 동일한 V8 엔진이 Chrome 브라우저에서 javascript를 실행하기 위해 Chrome 내부에서 사용됨
 - Google은 V8 엔진을 공개했으며, node.js의 빌더는 node.js에서 Javascript를 실행하는 데 사용
 - 현재 Node.js 바이너리는 V8 없이는 작동하지 않음 (예외 : Microsoft의 Chakra Javascript엔진)
- V8은 C++로 구성, 개발됨
 - 인터프리터를 사용하지 않고 자바 스크립트 코드를 직접 머신 코드로 변환

