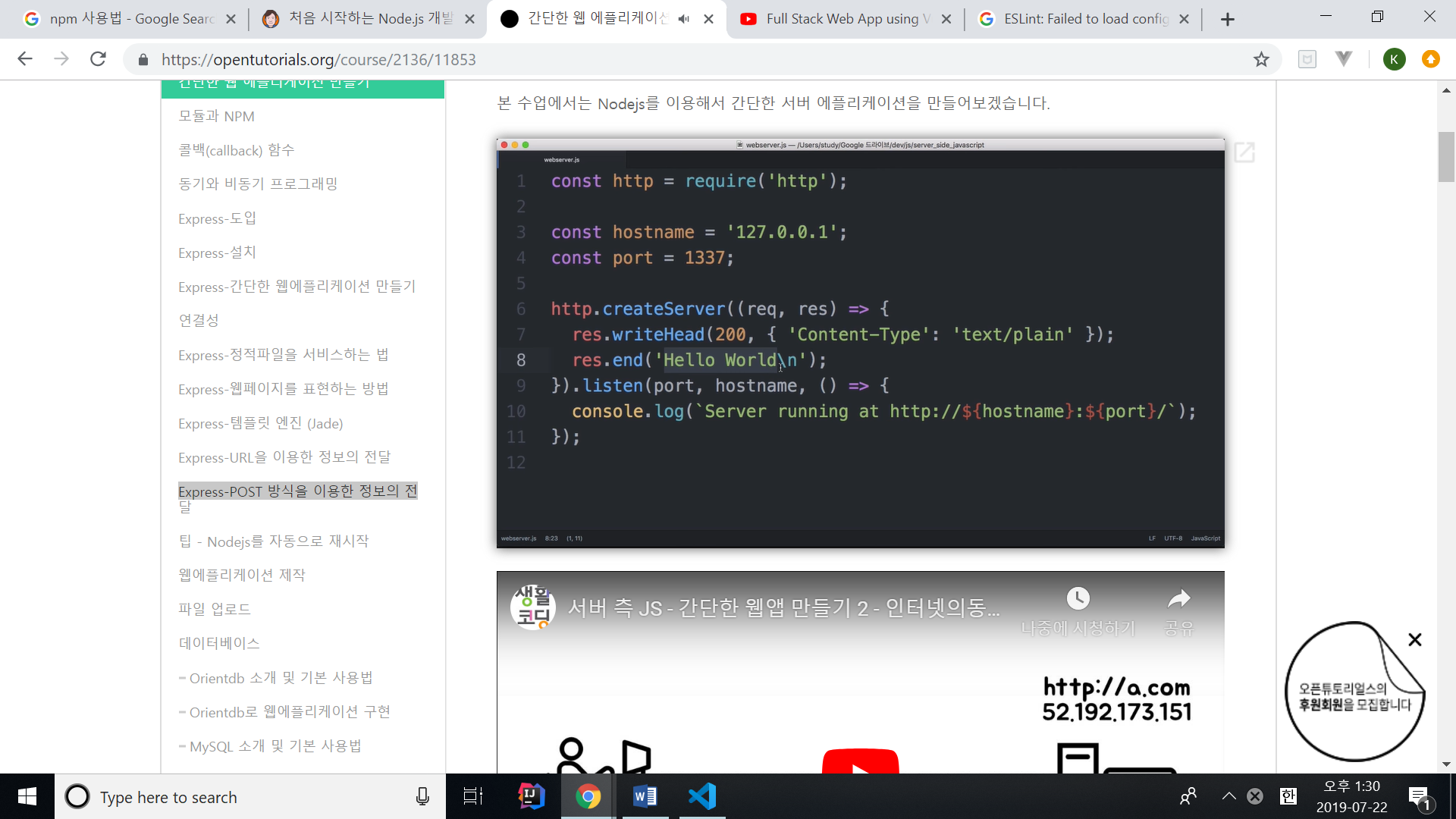
웹, 그리고 Node.js

1. 개념정리

**node.js**란 서버 사이드 서비스를 JavaScript로 구현할 수 있게 만든 런타임 또는 자바스크립트 언어를 통해 서버 쪽에서 동작하게 함, npm은 node.js 기반의 모듈을 모아둔 집합 저장소

**웹서버**는 클라이언트로부터 요청을 기다림. 요청이 들어오면 요청에 해당 정보를 리턴해주거나 행위를 함으로써 웹이 동작.

**서버, 클라이언트, 아이피, 도메인, 호스트, 포트**



(1) 우리 눈으로 보이는 과정: 사용자가 웹브라우저에 주소 값을 입력 예) a.com -> 클라이언트가 인터넷을 통해 a.com 서버에 접속하여 어떤 정보를 요청 -> a.com 서버는 요청한 정보를 응답

(2) 실제 일어나는 과정: 도메인인 a.com은 사람이 기억하기 쉽도록 만든 이름일 뿐(전화번호부 이름) 현실에서는 도메인과 일치하는 아이피 값(127.0.0.1)이 이용되는 것(실제 전화 번호)

(3) a.com에는 채팅 서버, 웹서버, 데이터베이스 서버 등 다양한 서버가 존재. 그렇다면 어떻게 클라이언트가 a.com이라는 주소 값을 요청했을 때, 클라이언트가 원하는 서버를 연결시킬 수 있을까? 정답: 컴퓨터에는 0~65535개의 포트가 존재. 80번 포트에 웹서버를 연결(80번 포트를 리스닝 하게 한다고 표현). a.com:80 요청시 80번 포트에서 리스닝 하고 있는 웹서버가 응답. 여기서 :80은 생략 가능함 왜냐하면 주소 값 앞에 있는 http가 웹이라는 걸 설명해 주고 있기에. http를 통해 접속한 경우 80을 쓰자고 이미 약속.

(4) 해당 이미지 설명: node.js를 통해 웹서버를 create 하고, 1337포트를 리스닝 하게 시키는 코드. 127.0.0.1로 접속 사용자에게 응답하라

**콜백함수**

sort라는 함수의 파라미터에 다른 함수(얘가 콜백)를 준다. 내가 콜백을 부른 것이 아니라 sort라는 함수가 내부적으로 필요할 때마다 콜백을 부름. 즉 콜백이랑 나중에 호출당할 함수.

2. 명령어 정리

npm init: npm 을 사용하겠다. (package.json 파일 생성)

npm init –f (질문없이 바로 시작)

npm install <패키지>: 필요한 패키지 add on. node\_modules 폴더가 생성됨

npm install -g nodemon 동일 프로젝트 파일에서 저장되는 것이 아니라 global로 저장하면 C:\Users\tkfkd\AppData\Roaming\npm\node\_modules\nodemon\bin\nodemon.js에 저장이 된다. 참고로 노드몬 패키지는 프로젝트 폴더의 파일들을 모니터링하고 있다가 파일이 수정될 경우 자동으로 서버를 리스타트 시켜줍니다.

npm install eslint --save

npm install --save express body-parser cors morgan

body-parser 미들웨어는 post로 요청했을 때 요청 파라미터를 확인 가능. 즉 html 문서에서 백엔드로 데이터를 보낼 때 사용자에 맞게 데이터를 추출하고 가공하는 일련의 과정이 파싱.

express는 웹서버를 생성.

cors: 먼저 CORS란 Cross Origin Resource Sharing의 약자로, 현재 도메인과 다른 도메인으로 리소스가 요청될 경우를 말한다. 예를 들어, 도메인 http://A.com 에서 읽어온 HTML페이지에서 다른 도메인 http://B.com/image.jpg를 요청하는 경우를 말한다. 이런 경우에 해당 리소스는 cross-origin HTTP 요청에 의해 요청된다. 보안 상의 이유로, 브라우저는 CORS를 제한. 그래서 이 모듈은 이를 가능하게 만든다.

morgan: 로그 기록을 남기는 morgan 모듈. 클라이언트의 HTTP 요청 정보를 로깅하기 위한 모듈

npm install –save axios (클라이언트 쪽에서)

axios란 Promise 기반의 자바스크립트 비동기 처리방식

npm install –save joi

node frame works to validate body request or query etc. POST API 등으로 신규 값을 추가할 시 전달된 값의 유효성을 검사

npm run lint -- --fix

참고 사이트: <https://november11tech.tistory.com/140>

3. Sequelize란 무엇인가?

Sequelize는 node.js 기반의 ORM으로 Promise 문법을 사용

먼저 ORM(Object Relational Mapping)을 알아야 함. 객체(Object)와 DB의 테이블이 일대일 대응하는 관계를 맺는 것. 데이터 베이스에 저장된 레코드를 객체로 바꿔 표현한다는 의미로써, 즉 ORM에서 객체 == 테이블

ORM을 이용하면 쿼리가 아닌 메서드로 데이터를 조작할 수 있음

예시) RDB(관계형 데이터 베이스)방법 select \* from user VS ORM 방법 user.findAll()

참고 사이트: https://jongmin92.github.io/2017/04/28/Node/sequelize/