S H O U T . O U R . P A S S I O N . T O G E T H E R

21st UNI SOPT

02.
Node.js 소개
기본 내장 모듈
HTTP
NPM, 외장모듈

SOPT

Contents

21st UNI SOPT

2nd Seminar

1. Node.js 소개

- Hello Node!
- non-blocking & 비동기
- Node.js의 장단점

2. 기본 내장 모듈

- url
- querystring
- crypto

3. HTTP

- http 소개
- Request
- Response

4. NPM, 외부 모듈

- NPM이란?
- NPM 사용법
- Request
- File System
- CSV

SOPT

2 차 세 미 나

01. Node.js 소개

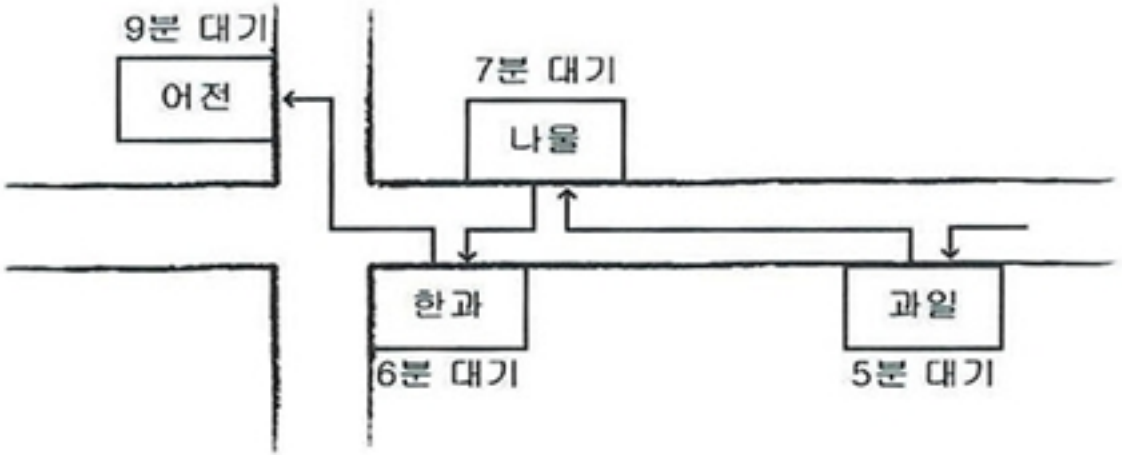
Hello Node!

- JavaScript 기반의 서버 플랫폼
- Google JavaScript Engine V8 기반
- 싱글 쓰레드의 non-blocking I/O 비동기 방식
- 이벤트 기반

SOPT

non-blocking & 비동기

싱글 쓰레드의 non-blocking 비동기방식???



시장에서 물건을 사는 상황을 가정.

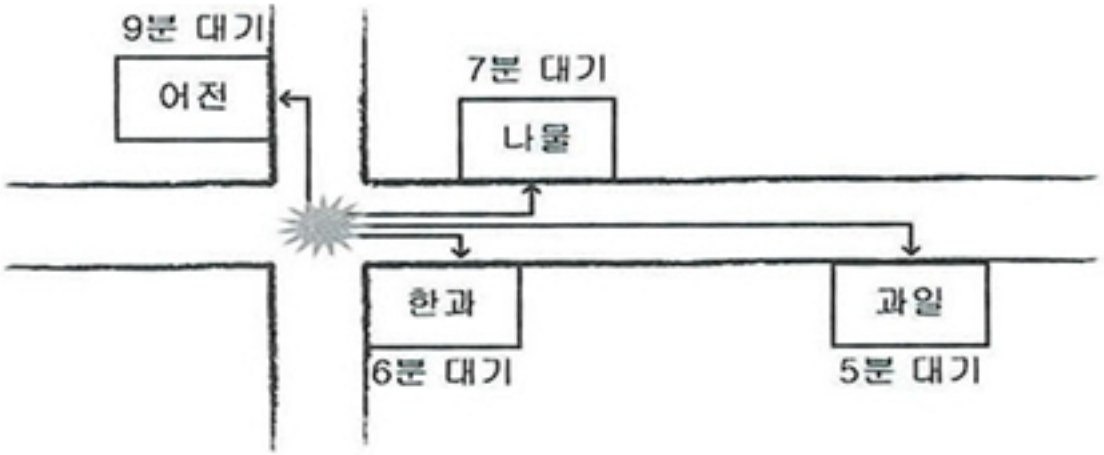
4개의 가게를 들러야 함

총 5 + 7 + 6 + 9 = 27분의 시간이 듦. (이동시간 무시)

시간을 단축시키는 방법은?

non-blocking & 비동기

방법 1) 4명이서 각각 1군데의 가게를 들린 후 다시 모인다.



가장 많은 시간이 걸리는 어전에 간애가 돌아올때까지 약 9분 소요.

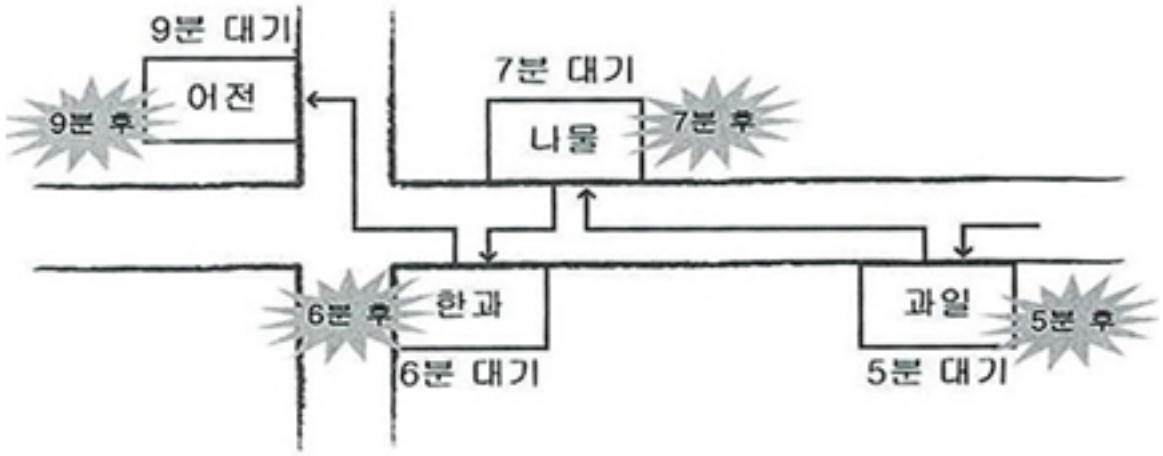
⇒ 동기방식, 쓰레드를 여러개 만들어서 동시에 일을 처리

PHP, ASP, Spring 등의 방식

쓰레드간 데이터 전달의 문제, 작업의 증가시 쓰레드 증가로 메모리 증가

non-blocking & 비동기

방법 2) 1명이 각 가게에서 주문 후 알림벨을 받아 다된 가게에서 물건을 가져온다.



가장 많은 시간이 걸리는 어전에서 물건을 가져오기까지 약 9분 소요.

⇒ 비동기방식, 한개의 쓰레드로 여러개의 일을 빠른시간에 요청 후 처리

Node.js의 방식

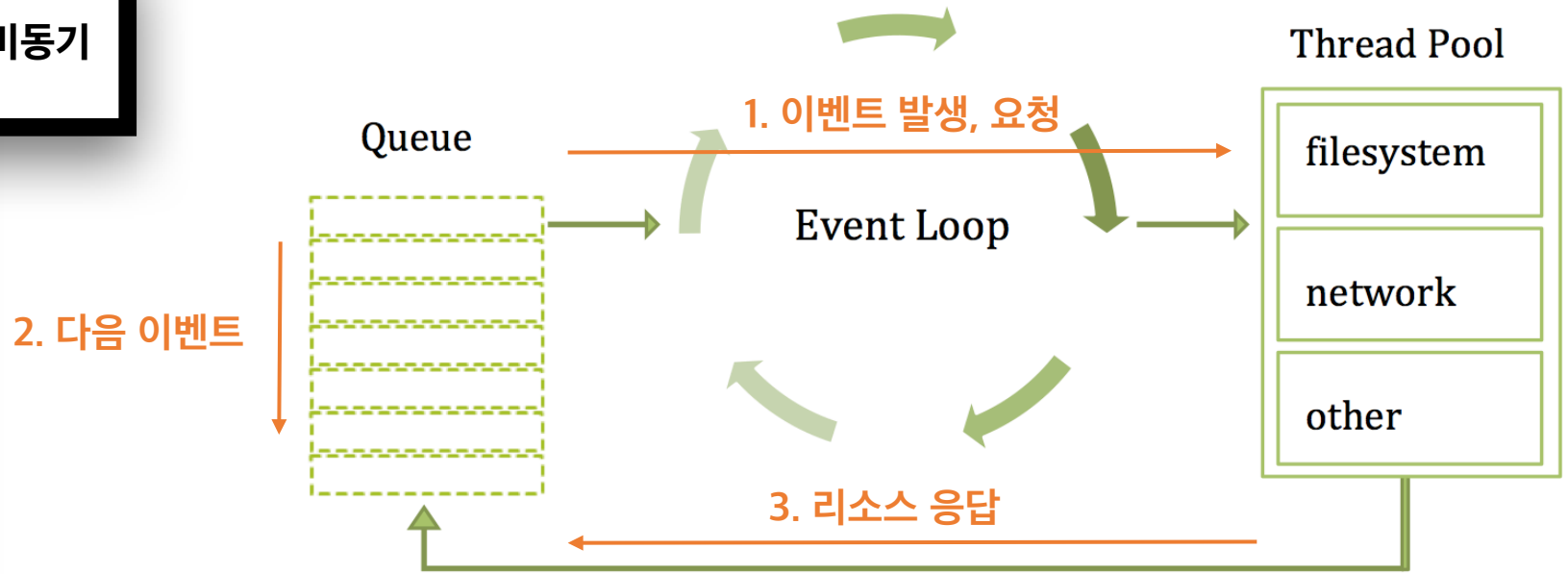
작업이 아무리 많아도 시스템 리소스의 변화가 없다.

-> 대규모 네트워크에 적합.

SOPT

Node.js 의 실제 동작방식

non-blocking & 비동기



이벤트 루프가 돌면서 이벤트로부터 요청을 받음

쓰레드 풀에서 요청을 처리

요청이 처리되면 리소스를 응답 받음(callback)

* 이벤트 루프는 자바스크립트의 특성이 아니다! 웹브라우저, 노드에서 동작한다.

Node.js의 장단점

Node.js 의 장점

- 싱글스레드, 비동기 I/O 처리로 빠른 속도
- 시스템 리소스의 부하가 적다.
- 프론트엔드 개발자의 진입장벽이 낮다.
- Spring, ASP등의 기존 서버사이드 프레임워크에 비해 생산성이 높다.
- 구글의 V8엔진기반으로 꾸준한 개발 및 지원이 예상된다.

Node.js 의 단점

- 싱글스레드이므로 큰 작업이 들어오면 부하가 크게 걸린다.
- 이론상으로는 빠른데 V8엔진의 성능이슈가 있다.
- 적절한 코드를 작성하지 못할 시 가독성이 크게 떨어진다.
- 에러가 발생하면 프로세스 자체가 죽는다.
- 개발진이 좀 자주 많이 크게 싸운다

대규모 프로젝트, 게임서버보다는 RESTful API서버, 채팅서버등에 적절!

SOPT

2 차 세 미 나

02. 기본 내장 모듈

SOPT

URL

- url 모듈 : url의 정보를 파싱할 때 사용하는 모듈
- 기본 내장 모듈
- url의 정보를 파싱하여 url 객체로 반환, 문자열로 반환 등
- 주요 메소드 :
  - parse(urlString)
  - format(urlObject)
- 그 외 메소드 : https://nodejs.org/dist/latest-v8.x/docs/api/url.html 참조

Query String

- querystring 모듈 : url의 query문을 파싱하는 모듈
- 기본 내장 모듈
- 쿼리문을 파싱하여 JSON 객체로 반환
- 주요 메소드 :
  - parse(queryString)
  - stringify(JSONObject)
- Query문이란?
  - HTTP에서 데이터를 전달하는 양식
  - key1=value1&key2=value2&.....
  - path와 query는 ? 로 구분
  - http://localhost:3000/test?key1=value1&key2=value2
- 그 외 메소드 : https://nodejs.org/dist/latest-v8.x/docs/api/querystring.html 참조

Crypto

- crypto 모듈 : 문자열을 암호화, 복호화, 해싱하는 모듈
- 기본 내장 모듈
- 비밀번호를 해싱하여 저장하는데 주로 사용
- 주요 메소드 :
  - createHash(algorithm)
  - update(string)
  - digest(encoding)
  - randomBytes(length, callback)
  - pbkdf2(string, salt, iterations, length, algorithm, callback)
- 그 외 메소드 : https://nodejs.org/dist/latest-v8.x/docs/api/crypto.html 참조

Crypto

단순 해싱하기

- crypto.createHash(algorithm)
  .update(string)
  .digest(encoding);
  의 방법으로 string을 해싱 가능하다.
- Hashing Algorithm : SHA256, SHA512, SHA1, MD5...
- 주로 SHA256, 최근엔 SHA512 많이 사용
- Encoding으로는 data64, hex등이 사용

- 레인보우 테이블을 가진 레인보우 공격에 취약하다!
- 실제로는 절대 이 방법만 사용해서는 안됨. 하나마나한 해싱

Crypto

보완방법

1. Salting

- Salt : 해싱을 할때 추가되는 바이트단위의 임의의 문자열
- Salt을 해싱할 문자열에 추가하여 해싱하는것을 Salting 이라 한다.

  salt + password string => digest
- 하나의 salt값을 코드에 지정해놓고 사용하면 역시나 소용없다!
- random생성된 salt값을 사용, DB에 salt값을 같이 저장.

Crypto

보완방법

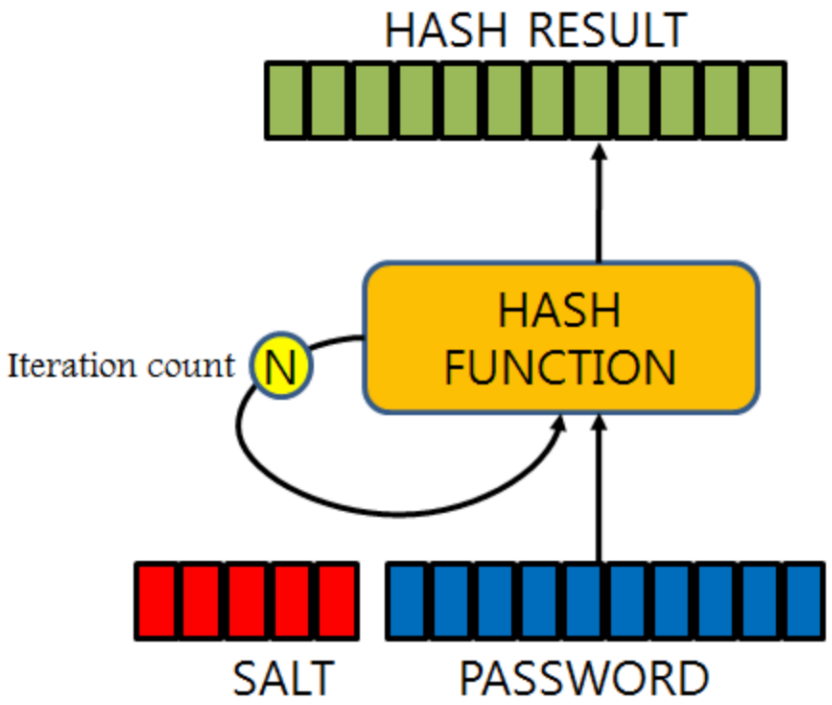
2. Key Stretching

- Hashing 함수를 반복하여 해싱하는 것.
- Salt값을 혹시나 알더라도 반복횟수가 일치하지 않으면 Digest를 찾을 수 없다.
- Hashing 하는데 시간을 오래 걸리게 함.
- 단순 해싱은 1초에 수만 ~ 수십억개의 Digest 생성가능

  -> GPU를 사용한 brute-force 공격에 쉽게 뚫린다!
- 사용자 기준으로 1초에 약 5회 정도의 Digest가 생성되도록 설정하는것이 좋은 설계.

Crypto

보완방법



Salting과 Key Stretching을 같이 사용하여 해싱하여야 한다!

Crypto

Salting + Key Stretching 을 이용한 해싱

- crypto.randomBytes(byteLength, callback)
  으로 랜덤 salt생성
- 콜백함수 안에서
  crypto.pbkdf2(string, randomBytes, iterations, length, algorithm, callback)
  으로 해싱.
- 이때 인자로 들어가는 randomBytes는 toString을 사용하여 인코딩해줘야 함.
- 해싱 알고리즘으로는 SHA256, SHA512 등을 선택
- length는 해싱된 문자열의 길이로 임의로 설정

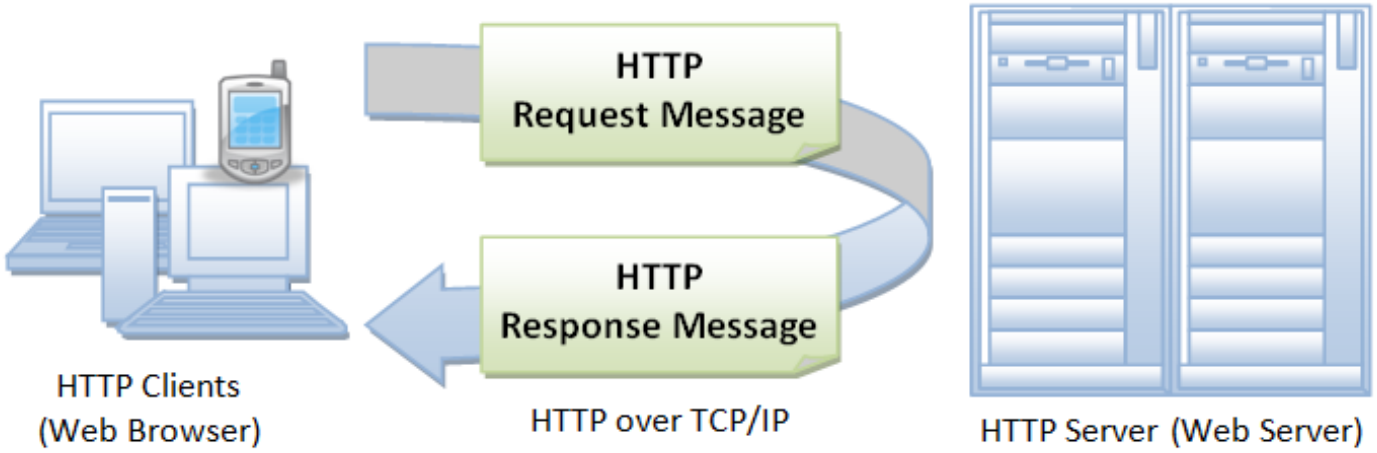
2 차 세 미 나

03. HTTP

HTTP 소개

HTTP : Hyper Text Transfer Protocol

- TCP, UDP를 사용하여 www에서 데이터(주로 하이퍼텍스트)를 주고받는 프로토콜
- 클라이언트와 서버 사이에서 request, response 가 이루어짐.



HTTP 소개

HTTP 모듈로 서버 열기

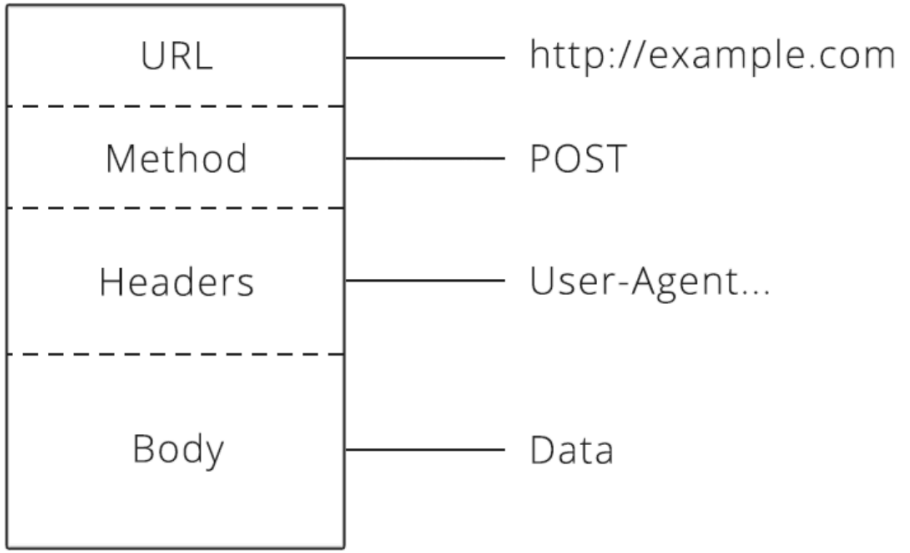
- http 모듈 추출하기
- 추출한 모듈에서 createServer(callback) 메소드를 사용하여 서버 생성
- 생성한 서버에서 listen(port) 메소드를 사용하여 서버 실행
- port는 자유롭게 지정할 수 있다.

0 ~ 1023	well-known port
1024 ~ 49151	registered port
49142 ~ 65535	dynamic port

- 사용하고자 하는 포트가 이미 사용중일수도 있으니 에러난다면 확인하기!

Request

HTTP Request 메세지 구조

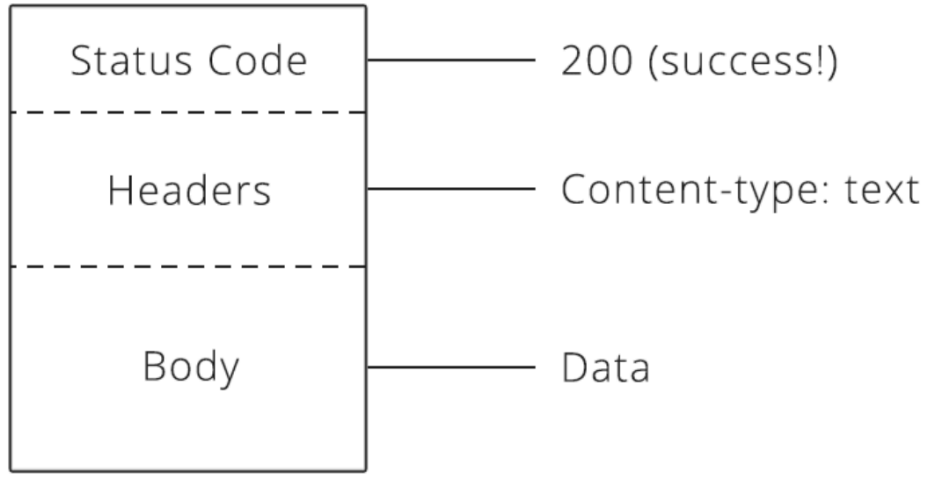


Request

- URL에는 요청할 url 주소가 들어감
- Method에는 GET, POST, PUT, DELETE 등의 통신 메소드가 들어감
- Headers에는 Content-Type, Content-Length, Host 등이 들어감
- Body에는 데이터가 html, json, xml, form-data 등의 형식으로 들어감

Response

HTTP Response 메세지 구조



Response

- 상태코드에는 1XX, 2XX, 3XX, 4XX, 5XX 등이 들어감
- Headers에는 Content-Type, Content-Length, Set-Cookie, Token 등이 들어감
- Body에는 데이터가 html, json, xml, form-data 등의 형식으로 들어감

SOPT

Response

주요 Status Code 종류

- 200 : 요청 전송 성공. (get 성공)
- 201 : 서버에 데이터 생성 성공 (post 성공)
- 304 : 리다이렉션 후에 리소스 수정된 거 없음. (오류 아님)
- 400 : request 오류403 : request 처리 거부
- 404 : 없는 리소스(URL)를 접근할 때 발생 (404 Not Found)
- 500 : Internal Server Error (서버에서 요청처리 불가)
- 501 : 서버가 지원하지 않는 요청을 한 경우 (서버에서 API가 구현이 안 된 경우)
- 503 : 서비스 자체가 불가능한 상태

Response

주요 Content-Type 종류

- application/xml : xml 데이터
- application/json : json 객체 데이터
- application/x-www-form-urlencoded : html from 데이터
- multipart/formed-data : xml 데이터
- text/plain : text 데이터
- text/html : html 데이터

Training 1

실습 1

1. 서버에 들어오는 url을 파싱하여 query로 만들고 이를 JSON 객체로 만들어주세요.
   query의 프로퍼티 키값은 str 입니다.
2. 서버에 접속시 str 값을 해싱합니다.
   * algorithm : SHA512, salting, key stretching을 모두 사용해주세요.
3. 해싱이 성공하면 JSON 데이터로 Response합니다.
   *JSON객체의 프로퍼티 키에는 msg, hashed가 있습니다.
   msg에는 성공 혹은 실패 메세지를, hashed에는 해싱된 문자열을 넣어주세요.
4. 3000번 포트로 해당 서버가 동작하도록 열어주세요.

코드는 이름_training2-1.js 로 저장하여 드라이브에 올려주세요!

SOPT

2 차 세 미 나

04. NPM, 외부 모듈

NPM이란?

NPM : Node Package Manager

- Node.js의 패키지 생태계
- Node.js의 오픈소스 모듈을 모아둔 저장소
- 세계에서 가장 큰 오픈소스 라이브러리 생태계
- 모듈의 버전관리가 쉽게 가능함
- Node.js가 빠르게 자리잡게 한 원동력!

NPM 사용법

명령어

- 패키지 설치 : npm install 모듈명

  ex) npm install ejs

  npm install fs

  npm install bcrypt 등등...

- npm install 만 입력시 package.json에 기록된 모든 패키지 설치

설치 옵션

- --save : 모듈을 설치하며 package.json 파일에 모듈정보 저장

  -> 다른사람의 환경에서도 npm install 로 프로젝트의 모듈 쉽게 설치가능

  ex) npm install --save http

  npm install mysql --save

- -g : 전역모드로 모듈 설치 -> 해당 프로젝트가 아닌 시스템에서 필요한 모듈

  mac, linux에서는 sudo명령어로 root권한을 줘야한다!

  ex) npm install -g express-generator

  sudo npm install pm2 -g

Request

- request 모듈 : 서버 내부에서 다른 서버로 request를 보낼 때 사용하는 모듈
- 설치 : npm install request
- 다른 서버에 request를 보내 데이터를 받아옴.
- 주요 메소드 :
  - request(option, function(error, response, data))
- 그 외 메소드 : https://www.npmjs.com/package/request 참조

File System

- File System 모듈 : 파일을 읽고 쓰는데 사용하는 모듈.
- 기본 내장 모듈
- 파일 읽기 쓰기, 디렉토리 생성, 열기
- 주요 메소드 :
  - readFile(path, encoding, callback)
  - readFileSync(path, encoding)
  - writeFile(path, data, encoding, callback)
  - writeFileSync(path, data, encoding)
- 동기, 비동기 방식 모두 지원, 동기방식은 메소드명뒤에 Sync 붙음.
- 그 외 메소드 : https://nodejs.org/dist/latest-v8.x/docs/api/fs.html 참조

CSV

CSV란?

- Comma Separated Values
- Comma로 Column 값들을 구분한 데이터 파일
- xlsx 파일 등 엑셀파일에서 저장, 읽기 가능

  *엑셀에서 열어서 편집시 encoding이 깨질 수 있습니다!
- 엑셀파일을 DB로 옮기거나 DB를 엑셀파일로 옮기는 것이 가능
- 파일이므로 File System을 이용해 읽기, 쓰기 수행

CSV

json2csv

- csv형식으로 저장하기 위해 JSON객체를 CSV형식으로 바꿔주는 모듈
- 설치 : npm install json2csv
- 주요 메소드 :
  - json2csv({data : JSON배열 , fields : column 배열})
- 컬럼 배열을 필드값으로 JSON 객체의 값이 하나의 row로 들어감

csvtojson

- csv파일을 읽어와 JSON 객체배열로 바꿔주는 모듈
- 설치 : npm install csvtojson
- csvtojson.Converter를 추출한 다음 이것으로 새 컨버터 객체를 만들어서 사용

Training 2

실습 2

1. 서버에 요청이 들어오면 서버는 52.78.124.103:3000/training/2nd 로 request를 보냅니다. 이때 method는 GET입니다.
2. response되는 JSON객체의 property중 data를 CSV로 저장합니다.
   *response되는 객체의 내용을 모르겠다면 여기까지 작성후 console.log로 객체의 프로퍼티값을 확인 후 진행합니다!
   *fields 값은 response의 프로퍼티를 확인해서 배열을 만들어야 합니다!
3. 저장에 성공하였다면 저장된 CSV 파일을 열어서 response를 전송합니다.
   *request 모듈을 사용하며 받은 응답이 저장된 변수가 있더라도 꼭 CSV파일을 다시 열어서 response를 보내주세요!

코드는 이름_training2-2.js 로 저장하여 드라이브에 올려주세요!

SOPT

과제

1. localhost:포트/test 의 url로 접속시 서버는 GET 메소드로 52.78.124.103:3000/homework/2nd 로 request를 보냅니다.
2. 이때 받은 response JSON object를 파싱해 시간값만 response하여 웹페이지에 띄웁니다. Content-Type은 자유입니다.
3. localhost:포트/info 의 url로 접속시 서버는 POST 메소드로 52.78.124.103:3000/homework/2nd 로 request를 보냅니다.
4. 이때 request를 보내는 body는 key값으로 name, phone을 가집니다. value는 자신의 이름과 휴대폰번호 '010-xxxx-xxxx'의 스트링입니다.
5. 이름과 휴대폰번호에 해당하는 데이터가 없으면 homework서버는 fail을 응답합니다.
6. 해당하는 데이터가 존재하면 console.log로 데이터를 확인 후 적절히 파싱하여 csv파일에 이름, 학교, 학과, 이메일, 해싱된 휴대폰번호를 저장합니다.
7. 저장성공을 알리는 메세지를 웹페이지에 띄웁니다.

10월 29일까지 이름_homework2.js 로 저장하여 드라이브에 올려주세요!

SOPT

수고하셨습니다!